NTNU – Trondheim
Norwegian University of
Science and Technology

Fleet Deployment & Disruption Management in RoRo Liner Shipping

Andreas Sola Fischer
Håkon Stavn Nokhart
Henrik Andre Olsen

Industrial Economics and Technology Management
Submission date: June 2015
Supervisor: Kjetil Fagerholt, IØT
Co-supervisor: Magnus Stålhane, IØT
Jørgen Rakke, WWL

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Problem description from the master's thesis agreement

This thesis will study a problem that is faced by fleet deployment planners every day due to the uncertainty involved in shipping, namely disruption management. Disruption management is the problem of finding the best possible solution to a situation that has occurred as a result of disruptions somewhere in the supply chain. This disruption forces the planners to modify an initial tactical plan, and the possible actions taken to do this may for instance be speeding up a vessel, chartering in a vessel (hire a vessel for a short time period), re-scheduling the vessel to other cargo assignments etc.

The project's objective is two-fold. The first objective is to develop a mathematical model, and efficient heuristic solution methods, to create fleet deployment plans for a Roll-on Roll-off liner shipping company. The second objective is to study how it is possible to reduce the effect of disruptions that occur during the execution of these plans. This may include both proactive and reactive approaches to disruption management. To evaluate the effect of these approaches it may be necessary to develop a simulation model for the shipping operations.

Preface

This is the thesis for our master's degree at the Norwegian University of Science and Technology, Department of Industrial Economics and Technology Management. It was written during the spring of 2015 in the course TIØ4905.

The thesis is written within the field of operations research. The presented findings are split into two parts: In Part I we continue on the work done in Fischer, Nokhart, and Olsen (2014), and present a mixed-integer programming model, an adaptive large neighborhood search heuristic and a rolling horizon heuristic as decision support tools for Roll-on Roll-off (RoRo) liner shipping companies to solve a fleet deployment problem. In Part II we consider uncertainty during the execution of fleet deployment plans. Here, we identify possible disruptive events in RoRo liner shipping, suggest several robustness strategies and a re-planning recovery procedure, and evaluate these in a developed simulation-optimization framework.

We would like to thank our supervisors Kjetil Fagerholt, Jørgen G. Rakke and Magnus Stålhane for suggestions, interesting discussions, and valuable feedback.

Andreas S. Fischer, Håkon S. Nokhart and Henrik A. Olsen

Trondheim, June 2015

Cover photo: Wiki Commons (2015)

Abstract

This master thesis consists of two parts. In Part I, the maritime *fleet deployment problem* for a major Roll-on Roll-off shipping case company is studied. The objective of the fleet deployment problem is to assign vessels in the company's fleet to voyages that must be sailed within given time windows and to cover monthly demand. In this thesis we introduce an approach where demand is modeled as actual cargoes, and different speeds can be assigned to both voyage and ballast sailings. In addition to a mixed-integer program, two heuristics are developed; an *adaptive large neighbourhood search* and a *rolling horizon heuristic*. These may help find satisfactory solutions within an appropriate amount of time.

The models are applied to realistic problem instances varying both in size and complexity. Results show that the mixed-integer programming model is not able to solve the largest and most complex instances within a time frame of 10,000 seconds. The rolling horizon heuristic performed excellent for most of the instances both in terms of solution time and quality, and can be used as a decision support tool for the case company when planning the fleet deployment.

In Part II, we study disruption management in Roll-on Roll-off liner shipping. We introduce uncertainty to the fleet deployment problem by adding disruptive events in a simulation framework. First, we identify possible events and their associated impacts. Then we propose several *robustness strategies* which may be used in planning. These are considered as proactive ways of handling disruptions.

The basic model and the robustness strategies are in a computational study evaluated in a simulation-optimization framework, where a *re-planning recovery procedure* serves as real-time disruption management. The presented results show that by adding robustness to deployment plans, the incurred operating costs and delays are significantly lower. The re-planning procedure further reduces delays and costs, although by performing many alterations to the plans during operation.

Sammendrag

Denne masteroppgaven består av to deler. I del I studeres flåtedisponeringsproblemet for et større Roll-on Roll-off shippingselskap. Flåtedisponeringsproblemet består av å tilordne skipene i selskapets flåte til seilaser med gitte tidsvinduer og å dekke månedlig etterspørsel. I denne oppgaven introduserer vi en tilnærming hvor etterspørsel modelleres som faktiske laster, og forskjellige seilehastigheter kan tilegnes til seilaser og seilinger mellom disse. I tillegg til et blandet heltallsprogram presenter vi to heuristikker: Et adaptivt nabolagssøk og en rullende horisont-heuristikk. Disse kan bidra til å finne gode løsninger til problemet innen rimelig tid.

Modellene er anvendt på realistiske probleminstanser som varierer i størrelse og kompleksitet. Resultatene våre viser at heltallsprogrammet ikke klarer å løse de større og mer komplekse instansene innen en tidsramme på 10 000 sekunder. Den rullende horisont-heuristikken gjør det bra både når det gjelder tid og løsningskvalitet på de fleste instansene, og kan med fordel brukes som et beslutningsstøtteverktøy for selskapet i planlegging av flåtedisponering.

I del II av oppgaven studerer vi avvikshåndtering i Roll-on Roll-off skipslinjefart. Vi introduserer usikkerhet til flåtedisponeringsproblemet ved å legge til forstyrrende hendelser i et simuleringsrammeverk. Først identifiserer vi mulige hendelser og deres tilhørende innvirkninger, før vi foreslår flere robusthetsstrategier som kan brukes i planleggingen. Vi ser på disse som forebyggende tiltak mot fremtidige mulige avvik.

Den grunnleggende modellen og robusthetsstrategiene er i en numerisk studie evaluert i simuleringsrammeverket. En replanleggingsprosedyre er her brukt til avvikshåndtering i sanntid gjennom simuleringene. Denne prosedyren har som mål å minimere kostnadene når et påført avvik skal håndteres. De presenterte resultatene viser at ved å legge til forebyggende tiltak og øke robustheten i initielle planer blir de påførte operasjonelle kostnadene og forsinkelsene vesentlig lavere. Replanleggingsprosedyren reduserer ytterligere forsinkelser og kostnader, på bekostning av å påføre mange endringer til planene.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ALNS	Adaptive Large Neighborhood Search. See Chapter 5.
AST	Adjust Sailing Times. Robustness strategy, see Section 12.2.
COMBINED	Combining AST, PRST and REA as a robustness strategy. See Section 12.5.
D	Total days delayed. See Chapter 14.
FDP	Fleet Deployment Problem. See Section 1.4.
MIP	Mixed-Integer Programming. See Section 1.4.
NN	No speed adjustments, No recovery re-planning procedure. Simulation test setting, see Chapter 14.
OC	Operating Cost. See Chapter 7.
ON	Open speeds, No recovery re-planning procedure. Simulation test setting, see Chapter 14.
OR	Open speeds and Recovery re-planning procedure is activated. Simulation test setting, see Chapter 14.
PRST	Penalize Risky Start Times. Robustness strategy, see Section 12.4.
REA	Reward Early Arrivals. Robustness strategy, see Section 12.3.
RHH	Rolling Horizon Heuristics. See Chapter 6.
RHH-R-MIP-SO	RHH routing and MIP speed optimization. See Section 7.5
RoRo	Roll-on roll-off. Term used to describe a certain type of ships where vehicles can be driven on and off. See Section 1.4.
RP	Recovery Procedure calls. See Chapter 14.
SC	Simulated Cost. See Chapter 14.
ST	Solution Time. See Chapter 7.
SRASO	Sequential Routing And Speed Optimization. See Section 7.5.

TEU	Twenty-foot Equivalent Unit. Unit based on the volume of a 20-foot-long container. Often used to describe the cargo capacity of container carriers.
UV	Unserviced Voyages. See Chapter 7.
VS	Voyage swaps. See Chapter 14.

Glossary

Balance Category	Demands are defined for each balance category. A voyage going from Europe (EU) to Asia (AS) through North America (NA), may consist of three balance categories; EU-NA, NA-AS and EU-NA. See Figure 4.4.
Berth	A ship's allocated space in a port (noun), or parking the ship at its allocated space (verb).
Bunker Consumption	A term for fuel consumption used in the shipping industry.
Capacity Class	Collection of decks with similar carrying capacities and heights.
Capacity Group	The leg between two ports in a voyage.
Cargo Transshipment	Shipping of cargoes to an intermediate destination, i.e. to a port that is not the cargo's final destination.
Disruption	The new conditions set by an occurred event cause changes to a current plan.
Forecasting Period	The second part of the sub-horizon in the RHH-algorithm where some selected constraints are relaxed.
Primary Period	The first part of the sub-horizon in the RHH-algorithm that has the same constraints as the exact model.
Recovery Action	An action to mitigate the impacts of a disruption, e.g. increase vessel speeds.
Recovery Strategy	A set of recovery actions or ways to re-plan the deployment in order to mitigate the impacts of a disruption.
Robustness Strategy	Measure to adjust the mathematical model defined by (4.1)-(4.30) to consider more robust solutions.
Sailing Ballast	Sailing without any cargoes or load on board the ship.

Scenario	A set of predefined events used in the simulations. Contains information regarding event types, when they occur and their associated impacts.
Slow Steaming	Operating ships at speeds much lower than their maximum cruising speeds. Widely used to save fuel and reduce operating costs.
Speed Profile	The different speed alternatives the vessels can take on.
Problem Instances	In this thesis, several problem instances from the case company have been solved. The problem instances consist of real planning data from the case company's operations. See Section 7.1 for a thorough description.
Trade Route	A sailing route with several port calls in sequence. See Section 4.1.
Voyage	Performing a journey of trade route one single time. See Figure 1.2.

Chapter 1

Introduction

Maritime transportation is the main network of distribution for international trade (Christiansen et al., 2007) and is recognized as a crucial facilitator in the expansion of the global economy. According to UNCTAD (2013), more than 80 % of the world trade measured by volume is carried by seagoing vessels. There has been a systematic growth of maritime freight transportation in the later years. For example, during the first decade of the new millennium, the total container ship capacity went up by 164 % (Christiansen et al., 2013). Rodrigue et al. (2013) points out globalization, technical improvements and economies of scale as causes for this significant growth. During the same period there has also been a substantial improvement in the utilization of the world fleet, in terms of tons carried per deadweight ton and ton-miles performed per deadweight ton (Christiansen et al., 2007). Figure 1.1 shows how seaborne trade historically has exceeded the growth in GDP, and there is no indication of any turn of this trend in the near future. However, the competition in the transportation industry has increased and the maritime transportation industry is no exception. In addition to strong competition, climate change and other environmental concerns have become important drivers towards more efficient and robust transportation (Hoff et al., 2010). Shipping involves major capital investments and the daily operating costs of a ship can be tens of thousands of dollars. Better fleet utilization can therefore give significant bottom line improvements.

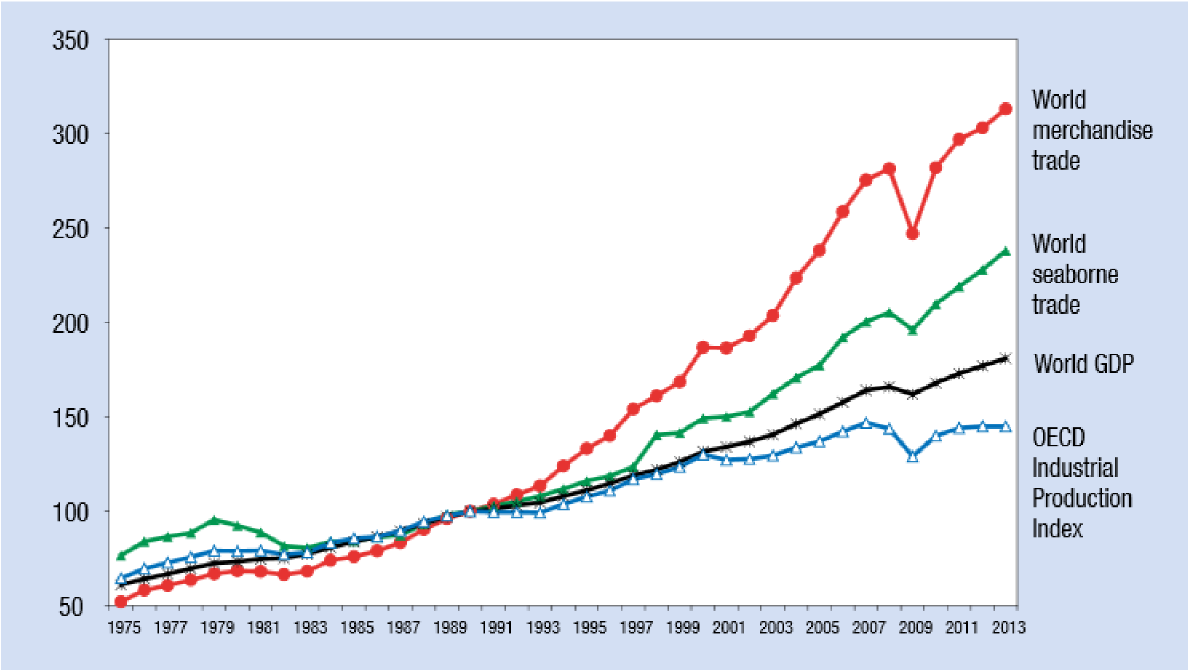

Figure 1.1 – The OECD industrial production index and indices for world gross domestic product, merchandise trade and seaborne shipments (1975–2013),(1990=100). Source: UNCTAD (2013)

Comparison to other freight transportation modes

When it comes to transporting large volumes of cargo among continents, there is no direct substitute for maritime transportation. Christiansen et al. (2007) mention pipelines as a possible substitute to parts of the shipping industry. However, limited to only being able to move fluids over fixed routes, pipelines is not a realistic alternative. In Christiansen et al. (2007), maritime transportation is compared to other modes of transportation like aircraft, truck and train, and the differences can easily be identified. Greater variety of vessel types, longer trip length of voyages and greater uncertainty are some of the most important characteristics that differs in maritime transportation from the other modes of freight transportation.

Maritime transportation has until recently been devoted little research attention compared to for example aircraft and road transportation. Christiansen et al. (2007) identify low visibility, high uncertainty, high variety of problems and a conservative industry as four reasons that can explain the low attention of research. In our search for relevant literature we observed an increase in publications when it comes to operations research applied in the shipping industry since Christiansen et al. (2007) was published.

1.1 Shipping Segments

Lawrence (1972) introduces three segments of maritime transportation: industrial, tramp and liner shipping. In industrial shipping, the vessels are controlled by cargo owners who aim to minimize the costs of shipping their cargoes. In tramp shipping, the ship scheduling is made with a mix of mandatory and optional cargoes available, while liners operate according to a published vessel route schedule. These three segments can be illustrated through a practical example. The industrial shipping can be seen as trucks that transport goods for an owner, tramp shipping are taxis that transport customers exactly where they want to go, and liner vessels are buses operating the public transportation network (Ronen, 1983). According to World Shipping Council (2009), the liner shipping segment transported around 60 % of the total value of global seaborne trade in 2007.

Within the three categories of shipping operations, Christiansen et al. (2004) focus on decisions made on the strategic, tactical and operational planning levels. On the strategic planning level, long term decisions like determining the market and trade selection, ship design, network and transportation system, fleet size and mix, and the port location, size, and design are made. The tactical problems include fleet deployment, ship routing and scheduling, inventory ship routing, ship management, and distribution of empty containers. On the operational planning level the day-to-day decisions are made, like determining the speed of the vessels, the ship loading, and the environmental routing (Christiansen et al., 2007).

Christiansen et al. (2004) emphasize that the main objective for all shipping companies, although the basic conditions such as fleet size and mix may differ, is to utilize their fleets optimally. The decisions, however, may be different. For industrial shipping and tramp shipping, the major decisions include route and schedule design, fleet size and mix, optimal assignment of cargoes to each ship, and the ship routing and scheduling. In liner shipping, decisions include route and schedule design, fleet size and mix, fleet deployment, and cargo booking. Within liner shipping, the first two aforementioned decisions are made on a strategic planning level, while the latter two will be made on a tactical and operational planning level, respectively. A decision on the operational planning level that is both frequent and of great importance for all shipping segments is what method to use in real-time disruption handling. These methods are called recovery strategies and will be explained in greater detail in Section 1.3.

1.2 The Fleet Deployment Problem & Roll-on Roll-off Shipping

The fleet deployment problem (FDP) is a well-studied problem, and is defined as a tactical problem within the liner shipping segment by Christiansen et al. (2004). Consider a provider of liner shipping services with a fleet of vessels at its disposal; the problem deals with optimally assigning these individual vessels to trade route voyages in the planning horizon. A trade route consists of a number of port calls in sequence serviced by a vessel. According to monthly demand and contractual agreements, each trade route has to be serviced with a certain frequency. A vessel completing one instance of a trade route once is denoted as a voyage. Trade routes are usually long distance maritime shipping routes, and are determined by liner shipping companies by analyzing demand. The planning horizon for FDP is typically from a few months and up to a year.

As presented earlier, liner companies operate schedules similar to public bus services. When announcing its trade routes and entering contractual agreements, the company commits to a supply of transportation services. This supply needs to satisfy voyage frequency and capacity requirements. In other words, the shipping company needs to provide a certain number of voyages with sufficient capacities on a monthly basis. Due to global trade imbalances, a ship may have to sail ballast to the start of its next assigned voyage. If the shipping company does not have sufficient fleet capacity, there are in some markets possible to charter spot vessels to perform certain voyages. Figure 1.2 illustrates trade routes, voyages and ballast sailing in between voyages.

An optimal fleet deployment can normally be seen as either (1) maximizing profit or (2) minimizing costs. This depends on the basic business structure of the company, especially contractual issues and the access to spot markets. Usually, shipping companies solve the fleet deployment problem manually, with only the aid of comprehensive planning experience and simple spreadsheets in the process (Fagerholt et al., 2009). For larger fleets and longer planning horizons, this is obviously a very difficult task, and finding the optimal solution is almost impossible without adapting some form of decision support system.

The case company in this thesis is a major player in the Roll-on Roll-off (RoRo) vehicle transport services segment of liner shipping. RoRo vessels constitute a significant share of the company's fleet, and these vessels are designed to allow cars, trucks and trains to be driven directly on board. Originally appearing as ferries, these vessels are used

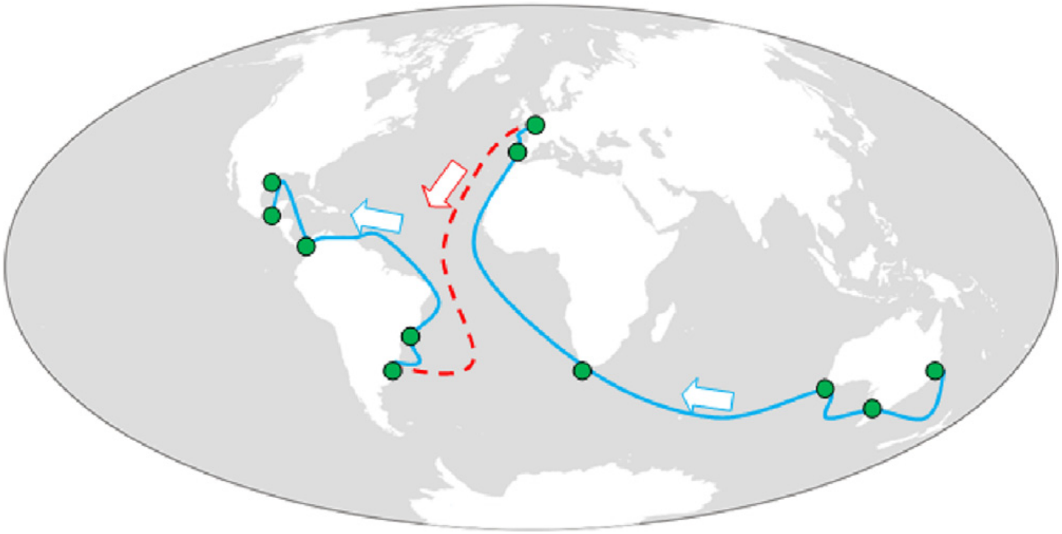

Figure 1.2 – Illustration of two successive voyages on two different trade routes, and with a ballast sailing in-between. Source: Andersson et al. (2014)

on deep-sea trade routes between continents and are much larger than a typical ferry. Their capacity is measured in the amount of parking space they are able to offer to the vehicles they carry, i.e. in lane meters. The data presented in Lindstad et al. (2012) shows that the RoRo segment in liner shipping is relatively small compared to the container shipping industry. This may explain why this segment has been devoted less attention in operations research, and the potential for substantial savings with better decision support may be present. In terms of revenues, the RoRo segment is indeed significant.

The design of RoRo vessels are rather complex. While containers can be stacked by cranes, vehicles must be driven aboard on *decks* inside the vessel. RoRo vessels usually have built-in ramps to efficiently allow cargoes to drive in and out while in port. Also, in contrast to containers with standardized dimensions, vehicles and equipment on wheels may come in all sizes and weights. This means that RoRo vessels must have a variety of decks available in order to carry different types of cargo. Therefore, special requirements to deck heights and carrying capacities are needed. Examples of products carried by RoRo vessels are regular cars, SUVs, trucks, farming equipment and military equipment.

1.3 Disruption Management In Liner Shipping

The fleet deployment problem is usually presented as a deterministic problem in the academic literature, i.e. parameter values are assumed known with certainty. However,

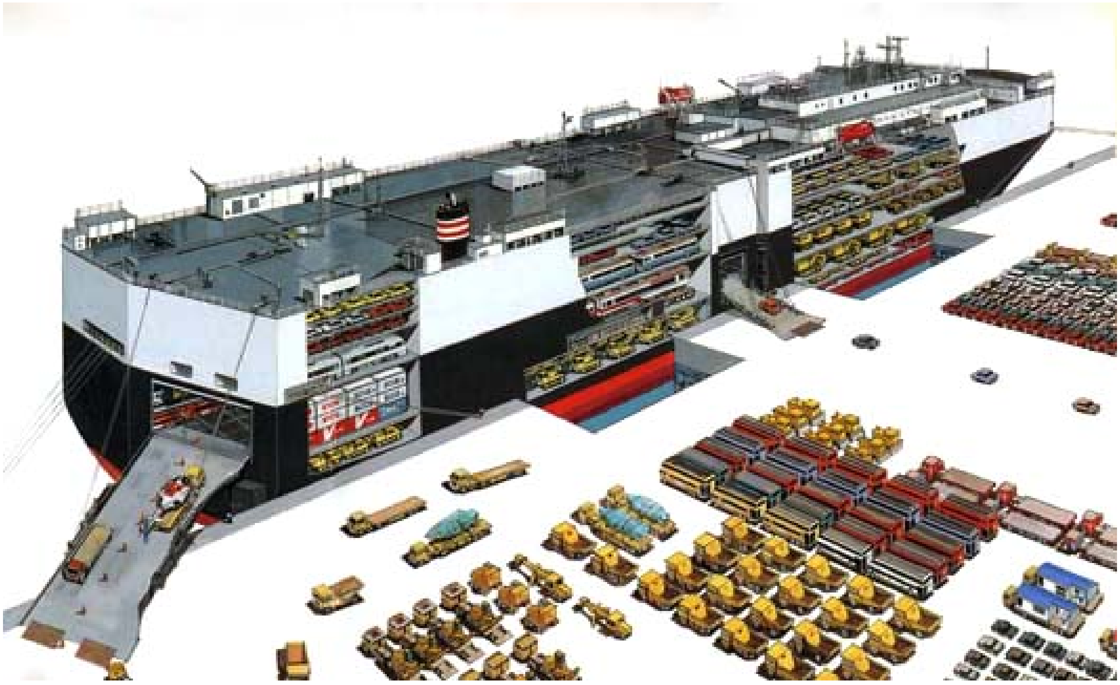

Figure 1.3 – Illustration of the loading/unloading of a RoRo vessel. Source: International Cargo/Export Shipping Co. (2014)

in reality maritime transportation operates in an environment that is highly uncertain and constantly changing. Hence, the execution of a predetermined fleet deployment plan are often subject to changes or *disruptions*. Disruptions are caused by some unforeseen *event*. Brouer et al. (2013) state that common events leading to disruptions in maritime transportation are strikes in ports, bad weather, congestion in passageways and mechanical failures. Other, but less frequent events include piracy and crew strikes on vessels. All types of events may lead to disruptions, such as delays or even infeasibility of the original plan. The occurrence of disruptions are common in a global shipping network. According to Notteboom (2006) only 20 % to 30 % of the global shipping lines showed schedule reliability. In other words, 70 % to 80 % of the lines were disrupted in some way. Even though most of the disruptive events only lead to minor delays, some events may have severe impacts on the shipping companies' operations. In January 2015, BBC (2015) reported that the car carrier Höegh Osaka developed a severe heeling shortly after sailing from Southampton. This forced the crew to beach the vessel on a bank nearby, which left the vessel out of service for an indefinite period of time, as shown in Figure 1.4.

Disruptions which cause vessels to deviate from the original plan could impose several costs for shipping companies (Brouer et al., 2013). First, voyage delays may incur addi-

Figure 1.4 – Picture of the car carrier Hoegh Osaka beached at a sandbank outside Southampton, Hampshire. Source: GettyImages

tional fees. Second, to minimize delays, shipping companies may speed up vessels, which significantly increases bunker consumption and thereby the operating costs. In addition, companies may rearrange vessel schedules, if possible, and accept its associated costs. We must also consider delays from a customer's perspective, i.e. demand for a reliable service and expectation of delivery on time. The costs of loss in credibility are difficult to measure, but they might be substantial in a competitive market. The handling of disruptions is therefore a major concern for shipping companies.

In the following section we introduce three important aspects when handling disruptions. First, the process of identifying different types of events is described. Then we explain the basics of pro-actively mitigating disruptions. Finally, we introduce concepts of real-time disruption management.

Identifying possible types of events

In order to efficiently handle disruptions, it is essential to identify the different types of events which may occur during shipping operations. Every event will have a unique impact and must be handled accordingly. The probability of different types of events varies and needs to be taken into account. Modeling the uncertainty requires insight in (1) the types of events, (2) the impact of types of events, and (3) a realistic probability for the occurrence of different events.

A robust fleet deployment model

The goal of robust planning is to make the vessel schedules less sensitive to disruptions. The main idea is to incorporate ways of handling disruptions in the original plan. Hence, robustness can be seen as a pro-active way of handling disruptions.

Clausen et al. (2010) divide robustness into two categories. The first category is *absorption robustness*. The aim of absorption robustness is to ensure plans remain feasible and that knock-on effects to the network is avoided. The most apparent way of adding absorption robustness when planning is to include time buffers in the vessel schedules. However, Clausen et al. (2010) point out that incorporating buffers leads to a trade-off between costs and robustness: Larger buffers will make a schedule more robust, but will initially appear more costly. In addition to the optimal trade-off between robustness and costs, (1) where to place the buffers and (2) the optimal amount of buffer time must be determined.

Absorption robustness may not be sufficient when major disruptions occur. In order to restore operations in these situations, recovery robustness, which is the other type of robustness, must be considered (Clausen et al., 2010). Here, the purpose is to design plans that fit well with existing *recovery strategies*. Recovery strategies are sets of actions shipping companies may use to recover from disruptions. These actions can be speeding up vessels, swapping voyage assignments or reallocating cargo, for instance, and are further described in the next subsection. A plan designed with emphasis on recovery robustness will more likely be able to make use of recovery actions to mitigate the impacts of events. These plans are still more expensive than cost-optimal plans. Again, we have a trade-off between costs and robustness.

Recovery strategies

Clausen et al. (2010) define recovery strategies as methods to generate recovery plans in the event of disruptions. These strategies are used when the original plans are found infeasible or apparently of poor quality. The generation of recovery plans is a complex operation, since a small alteration of the plan usually will lead to a re-planning of other larger parts of the plan. In the event of a disruption, it is common to solve the re-planning problem in a prioritized sequence with respect to the problem components. First, infeasibility in the vessel schedules are resolved. Then, the solution is improved with respect to costs.

As stated in Clausen et al. (2001), disruptions may lead to severe delays in the ongoing operations. Since delays are costly, the recovery problem needs to be solved as quickly as possible. This means that it is essential to develop recovery strategies that find good solutions to the recovery problems in a short amount of time. Regardless, it is apparent that the more time that is given to solve the recovery problems, the better are the resulting solutions. Hence, there is a trade-off between finding good solutions and solving the problem quickly.

1.4 Scope and Area of Studies

This master thesis consists of two parts. In Part I, a mathematical formulation to the FDP for RoRo shipping companies is presented. In order to find good solutions within a reasonable amount of time, two heuristic approaches are presented and compared to the use of a commercial solver on a Mixed-Integer Programming (MIP) model. In Part II we introduce uncertainty to the FDP by adding disruptive events and simulate over the planning horizon. First, we present several robustness strategies with the purpose of creating more robust plans. Then, we describe how the developed simulation framework may introduce disruptive events, and how we test and compare plans generated with alternative robustness strategies. We also show how recovery strategies could improve the performance of plans.

The following sections will briefly introduce Part I and Part II and set the scope of the studies.

Part I: The Fleet Deployment Problem in Roll-on Roll-off Liner Shipping

The objective of this part is to solve the RoRo liner shipping fleet deployment problem described in Section 1.2. In this part the parameters of the problem are considered known through the entire planning period and the operations are not subject to any kind of disruptions.

The mathematical model given in Chapter 4 is designed to find the optimal fleet deployment plan when minimizing costs. Through the decision variables presented, the model will allocate the vessels to voyages on the trade routes predefined by the shipping company. To make the model more realistic we integrate separate speed variables for the voyage and ballast sailings, as opposed to what is done in previous literature. This provides more flexibility when solving the problem, and thus may reduce the operating costs. In addition, to capture the variety of products and deck capacities on vessels in the RoRo segment, the cargoes and different deck types are segmented and specifically formulated.

When expanding the model to handle RoRo cargo and speeds as realistic as possible, the number of variables grows significantly. Even though new restrictions are needed, the solution space still becomes too large to find optimal solutions within a reasonable amount of time with commercial optimization software. In other words, implementing a realistic mathematical model that solves to optimum has no or little practical value for the case company due to the computational time required. However, solving large instances of the FDP through the use of heuristics may be possible. The results obtained by Andersson et al. (2014) suggest this, as the authors point out the need for an heuristic approach for longer planning horizons. Because the mathematical formulation presented in this thesis is even more complicated, it is reasonable to assume the same results. The presented *adaptive large neighborhood search* and *rolling horizon* heuristics can help to find good solutions to the problem in considerably shorter computing time.

In Part I of this thesis, Chapter 2 present a more thorough problem description to the FDP for RoRo liner shipping companies. In Chapter 3, previous relevant literature for this problem are reviewed. In Chapter 4 we present a mathematical formulation to the problem which forms the basis of the developed MIP. In Chapters 5 and 6, an adaptive large neighborhood search (ALNS) and a rolling horizon heuristic (RHH) are presented. With both these heuristics we aim to solve the FDP within considerable shorter time than the MIP, and thereby they may serve as more applicable decision support tools. A computational study of applying the MIP, ALNS and RHH on five different problem instances is given in Chapter 7. Finally, in Chapter 8, the main results are summarized and briefly discussed, and future areas of study are suggested.

Part II: Disruption Management in Roll-on Roll-off Liner Shipping

In this part we introduce uncertainty to the parameters in the problem presented in Part I. This is done by simulating the planning horizon and adding event scenarios which may affect the performance of the plan. The aim behind this procedure is to mimic the execution of real life shipping operations, and test actions to mitigate disruptions on both the tactical and operational level of the problem.

On the tactical level, we develop a set of robustness strategies which may help plans to be better prepared for future disruptions. The robustness strategies are variations of adding absorption robustness, recovery robustness and combinations of these, when generating initial plans. We evaluate the different robustness strategies in a simulation framework in terms of planned costs, incurred costs and delays.

On the operational level, we evaluate how the use of recovery strategies can help mitigate the impact of disruptions. When simulating initial plans in the developed framework, we evaluate (1) when recovery actions should be triggered, (2) what degree of actions which is the most efficient and (3) how they perform together with different initial solutions, i.e. fleet deployment plans generated without or with various robustness strategies. The recovery strategies consist of recovery actions which are designed to find new, less costly plans and to make a plan feasible again. Since the recovery procedure is used to resolve the problem during the execution of the shipping operations, the problem must be solved in a short amount of time.

In Part II, Chapter 9 gives a problem description of disruption management in RoRo liner shipping. Then we review relevant literature in Chapter 10. In Chapter 11, we identify events and their possible impacts on the shipping operations, and classify them as (1) sailing events and (2) port events. In Chapter 12, several strategies of adding robustness to fleet deployment plans are presented. How we allow for recovery actions and the simulation-optimization framework we use to evaluate solutions, robustness strategies and recovery strategies are described in Chapter 13. Then, we present a computational study based on this simulation-optimization framework in Chapter 14. Here, we evaluate which combination of robustness measures and recovery actions that produce the most cost-efficient and reliable solutions. In Chapter 15, our conclusions are summarized and possible future areas of studies are suggested.

Part I

The Fleet Deployment Problem in Roll-on Roll-off Liner Shipping

Chapter 2

Problem Description I

Part I of this thesis deals with solving the Fleet Deployment Problem (FDP) for a liner shipping company operating in the Roll-on Roll-off (RoRo) cargo segment. When RoRo liner shipping companies face the FDP, multiple decisions have to be made. These decisions vary in scope and importance, and below we describe each of these decisions and other considerations a RoRo liner shipping company must take into account when deploying their fleet with respect to minimizing costs.

Fleet Deployment Problem

The case company has a number of contracted trade routes that must be serviced a certain number of times each month. Trade routes consist of loading and discharging ports in different geographical areas. We denote one individual sailing of a trade route as a *voyage*. An illustration of trade routes and voyages is given in Figure 1.2. To meet demand and contractual obligations, a certain amount of voyages have to be carried out on the trade routes within each month.

In this thesis we assume there are two ways to cover contracted voyages: (1) either by using the shipping company's own vessels, or (2) by hiring vessels from external shipping companies. In reality, the option of hiring vessels from a spot market does not exist or is not sufficiently large in the RoRo segment to be a realistic alternative for the case company. However, to ensure feasibility of the model, the alternative of not servicing voyages is enabled. We denote voyages not serviced by vessels in the case company's fleet as unserviced voyages. As described further in Chapter 4, this option is associated with a substantial cost.

The contracted voyages have defined time windows in which the voyage must be started by a vessel. If a voyage is started later than the end of its time window, a delay cost is imposed. However, there is no preferences within the time window, meaning that there is no extra cost of starting a voyage in the later part instead of starting in the beginning of the time window. During the planning period, a vessel may have to sail several voyages, and a voyage does not necessarily start in the same port as where its preceding voyage ends. This may impose the vessel to reposition itself without any cargo before taking on its next voyage. We denote this as sailing ballast, and this should be held to a minimum. There are no ballast sailings associated with unserviced voyages.

The shipping company controls a heterogeneous fleet of vessels with different cargo capacities, sailing speed options and bunker consumption profiles. Since the vessels first have to finish any possible on-going voyages, they are available for new voyages at different times and in different ports. Due to port and passageway restrictions, among others, not all vessels are able to serve every voyage. In addition, as a requirement from the case company, some of the vessels may need to be dry-docked for repairs and maintenance during the planning period. This will keep the vessels out of service for a given period of time. These conditions confine which vessels that are suitable for taking on which voyages, particularly in the beginning of the planning horizon.

Further, the vessels have different deck setups and deck capacities (see Figure 4.3). The vessels' decks may take one or several types of cargo, and for each voyage we must decide which cargo to be placed on what deck. This may also affect which vessel we assign to voyages. For instance, for some trade routes the demand for one type of cargo might be severe, which requires a vessel with a large capacity of the deck(s) able to carry the given type of cargo. If the vessels sailing the voyages are not able to transport the total monthly cargo demand, the company has the option of hiring capacity at other liner shipping companies. We denote this as space chartering cargo. Unlike the non-existing spot market of vessels to cover voyages, the option of space chartering cargo is a realistic option to our case company and has a corresponding cost per m^3 of cargo transported.

When solving the fleet deployment problem in this thesis, we integrate the tactical planning, i.e. assigning vessels to voyages in the planning horizon, with speed optimization of the vessels. This means we assign speeds to vessels while considering the resulting operating cost and voyage duration. The speeds for sailing voyages and ballast sailing are considered separately.

Summary

The FDP entail a wide scope of costs that may occur when operating a large fleet of vessels in the RoRo liner segment. Operating costs depend on distance, speeds and other vessel characteristics. The objective when solving the FDP is to find the optimal fleet deployment with respect to minimizing costs. We can sum this up to deciding (1) which voyages and ballast sailings to be sailed by which vessel, (2) at what speed, (3) at what time, (4) and in which sequence. Also, decisions related to (5) cargo placement, (6) space charter and (7) unserviced voyages, if any, must be made. When making decisions, we must make sure voyages are serviced within their respective time limits, and that the monthly demand on the trade routes is met.

Chapter 3

Literature Review I

In this chapter we provide a thorough review of the relevant literature published within the field of operations research and maritime transportation. In Section 3.1 we present the development of studies on the FDP, before specifically addressing the relevant publications within the RoRo liner shipping segment in Section 3.2. This literature review was first conducted as a part of Fischer, Nokhart, and Olsen (2014), and is revisited here.

3.1 Fleet Deployment in Liner Shipping

The term *fleet deployment* is defined by Perakis and Jaramillo (1991, p.187) as the "...allocation of ships to routes, their general scheduling, and the chartering of vessels, if any, to complement the owned fleet in the fulfillment of the transportation missions". The same authors emphasize in their following paper, Jaramillo and Perakis (1991), that the FDP focuses on the tactical planning assuming that the strategic decisions have been made. In other words, decisions related to the fleet size and mix and the sequence of ports that forms each route are made at the strategic level and outside the scope of fleet deployment. Fagerholt et al. (2009) describes the FDP as determining an optimal way of servicing voyages defined for the planning horizon with the shipping company's fleet of vessels. Here, an optimal deployment is seen with respect to minimizing the costs. These costs typically consist of operating costs, voyage costs, maintenance costs and capital costs (Stopford, 2009). Further, Powell and Perakis (1997) studied the liner shipping industry and presented an integer programming (IP) model that is used to optimize the fleet deployment for a liner shipping company. The focus of the model is to minimize the operating costs for an existing fleet, given routing, ship availability, and service constraints. The results presented by the authors from applying the model were rather positive, which is shown in their comparison of their IP model against the original fleet deployment of the liner shipping company. The example shows a potential cost reduction of 1.5 %. When applying their IP optimization model in the fleet deployment for a liner shipping company, substantial savings were achieved.

Container shipping is a major segment of liner shipping, and a great deal of literature has been published in the recent decade to address the FDP faced by container liner shipping companies. Gelareh and Meng (2010) present a nonlinear mathematical model with speed optimization, which is linearized as a mixed integer linear programming model and solved for randomly generated numerical instances. The program suggests the optimal frequency pattern for each route operated by the liner company. In Meng and Wang (2010), the authors present a chance constrained programming model to solve the short-term liner ship fleet planning problem with cargo demand uncertainty for a liner container company. The demand uncertainty is modeled with a normal distribution, and for each liner route operated by the company, the chance constraints ensure that a minimum level of service is associated with each route. Wang et al. (2013) revisit the problem discussed in Meng and Wang (2010), but suggest that (1) an independent normal distribution of demand for each port pair, and (2) that all ships have to be empty at the start of a voyage, are not necessarily realistic modeling. Therefore, Wang et al. (2013) propose a joint chance constrained programming model, and show that the service level provided has significant effect on the total cost. Liu et al. (2011) formulate a joint optimization model for the container flow and fleet deployment problems. Finally, Wang and Meng (2012) present a cost minimization model for the FDP with container transshipment operations. The model allows containers to be delivered from its origin port to its destination port by the use of more than one single vessel.

3.2 Addressing the Roll-on Roll-off Segment

Andersson et al. (2014) point out that models presented by most of the aforementioned literature regarding liner container shipping is based on two assumptions: (1) a vessel that is allocated to a trade will operate on that trade during the whole planning horizon, and (2) each vessel is not treated individually, since a number of each vessel type is allocated. When considering fleet deployment within container liner shipping, these assumptions are generally applicable. For other important liner segments, such as RoRo vehicle transportation services, these assumptions can be too limiting. Fagerholt et al. (2009) suggest a new mathematical formulation for the FDP which do not rely on these assumptions. In addition, the authors present a multi-start local search heuristic approach to solve the FDP. Applying the model to an instance based on a RoRo case company gave results that indicated 2 - 10 % improvements compared to the solutions obtained from manual planning. Speed optimization is not considered in the model.

Unlike Fagerholt et al. (2009), integrated speed optimization and fleet deployment is studied in Andersson et al. (2014). Normally, a given service speed is assumed for each vessel when the planning is performed, while the speed optimization of the actual sailing speeds is made along the routes during operation. Andersson et al. (2014) emphasize that integrating speed decisions have been done in the industrial and tramp shipping modes, but there is a lack of literature within this field in liner shipping. Subsequently, the authors propose an integrated speed optimization and fleet deployment approach, and present a computational study on a real deployment problem in RoRo shipping. With this formulation, it is possible to determine the optimal speed for sailing a voyage and the subsequent ballast sailing to the next voyage. The formulation is simplified by having one common speed weight variable for these two speed decisions. They also present a rolling horizon heuristic for solving the problem, as the mixed-integer programming model is observed to only be solvable within a reasonable time for real instances with short planning horizons (Andersson et al., 2014).

The RoRo segment of liner shipping is small compared to container shipping, which may imply why the literature has been focusing on the latter. The total market size difference provides container shipping companies with a competitive advantage in terms of economies of scale and other synergy effects. However, even though some vehicles may be transported in containers as a substitute, the majority of vehicles shipped overseas is done through RoRo shipping (Øvstebø et al., 2011). Further on, Øvstebø et al. (2011) highlight how this part of the shipping industry must continuously improve in order to maintain its position, and present an optimization model for stowage planning for RoRo vessels at the operational level. The model provides great insight in how different cargoes in the RoRo segment differ in height, length, width and weight compared to standardized container dimensions in container shipping, and how cargoes carefully must be stowed at the different decks of the vessels. Even though this model deals with cargo stowage at the operational level, which is outside the scope of the FDP, it shows that demand and capacity issues in the RoRo segment must be modeled specifically in order for decision support tools to provide feasible solutions. However, this has not been included in the fleet deployment models presented by Fagerholt et al. (2009) and Andersson et al. (2014). We have not been able to find any literature that solves the FDP for RoRo shipping companies where demand is modeled as actual cargoes and vessel capacity is segmented into deck capacities.

Chapter 4

Mathematical Model

In this chapter we present a mixed-integer program to solve the fleet deployment problem stated in Chapter 2. In Section 4.1, the mathematical model is thoroughly described. Then, the formulations forming the MIP is given in Section 4.2, before giving a few formulation remarks in Section 4.3

4.1 Model Description

Let $\mathcal{V}$ be the set of vessels in the company's heterogeneous fleet, indexed by v. Each vessel has a set of discrete sailing speed options, $\mathcal{S}_v$, which it can choose from when sailing. Any convex combination of the speed options $s \in \mathcal{S}_v$ may be chosen for vessel v on a given leg, and the respective bunker consumption will be an overestimation of the actual consumption. This has been proven by Andersson et al. (2014) and is illustrated in Figure 4.1, where the piecewise linearizations at all times is above the consumption profile curve.

The set $\mathcal{R}$ consists of the trade routes operated by the company, and are indexed by r. The set of months in the planning horizon, $\mathcal{M}$, are indexed by m. $\mathcal{I}_{rm}=\{n_{r,(m-1)} + 1, n_{r,(m-1)} + 2, ..., n_{r,(m-1)} + \ n_{rm}\}$ denote the set of voyages on trade route $r \in \mathcal{R}$ starting in month m, and are indexed by i. Here, n_{rm} is the number of contracted voyages on trade route r in month m. Because the planning horizon starts at $m = 1$, we must define $n_{r,0} = 0, \forall r \in \mathcal{R}$.

This implies that a specific voyage is identified by its relevant trade route r and the voyage number i on this trade route. This can be interpreted as a *node* in a network consisting of

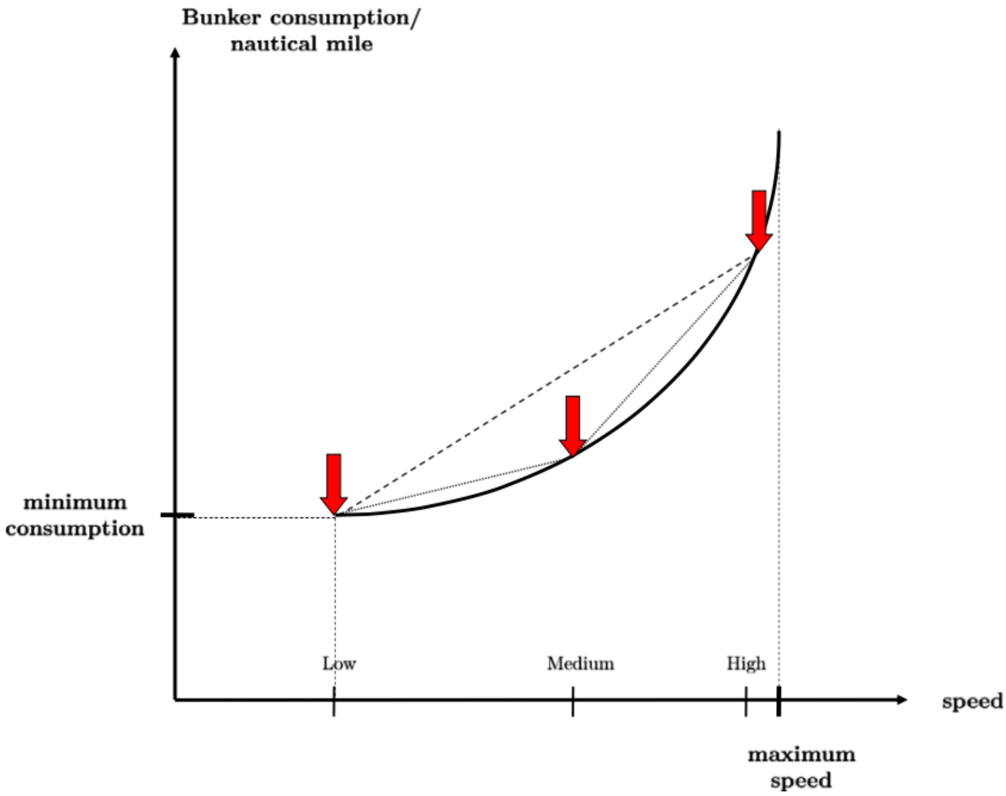

Figure 4.1 – Illustration of how the piecewice linearization of the fuel consumption is an overestimation of the actual consumption. Source: Andersson et al. (2014)

possible voyages. Every node in the model is associated with a combination of (r, i). Let $\mathcal{N}$ be the set of nodes. Each vessel in the fleet will start in an individual origin node that represents the position the vessel is available from. In order for a vessel to sail a voyage, it must first sail ballast from its origin node to the start port of the voyage. Each vessel will also sail to a destination node at the end of the planning period. This will mark when the vessel's sailing is terminated for the planning period. The artificial destination, as its name implies, does not exist physically, and will in reality be the same position as the final port of the last voyage sailed. In addition to voyages, $\mathcal{N}$ includes the origin node for each vessel, the artificial destination node for each vessel and maintenance nodes. The set $\mathcal{N}^C$ is a subset of $\mathcal{N}$ and contains the contracted voyages. $\mathcal{N}_v$ is also a subset of $\mathcal{N}$, and corresponds to the set of voyages that vessel v can service. Let $o(v) \in \mathcal{N}_v$ and $d(v) \in \mathcal{N}_v$ be the origin and destination nodes, respectively, for each vessel $v \in \mathcal{V}$. For $d(v)$, the distance from any node in $\mathcal{N}_v$ to $d(v)$ is zero. Just as the voyages, the maintenance nodes have two indices (r, i), and the set $\mathcal{N}_v^M$ is the set of required maintenance nodes for vessel v. For all but the vessels which are due for dry dock maintenance, the set $\mathcal{N}_v^M$ will be empty.

In addition to a network of nodes, we need to define a set of arcs to connect the nodes. Let an arc $((r,i),(q,j)) \in \mathcal{A}$ represent sailing ballast after visiting voyage or maintenance node (r,i) to the start of voyage or maintenance node (q,j). The set $\mathcal{A}$ also includes the arcs from the origin nodes to voyage and maintenance nodes, $(o(v),(r,i))$ and the arcs from the voyage and maintenance nodes to the ending positions, $((r,i),d(v))$. Similar to the node subset, we can define $\mathcal{A}_v$ as the subset of arcs that vessel v can traverse. By using this definition of arcs and nodes, the problem can be seen as defined on a graph $G(\mathcal{N},\mathcal{A})$. Figure 4.2 illustrates how arcs and nodes are connected in the model.

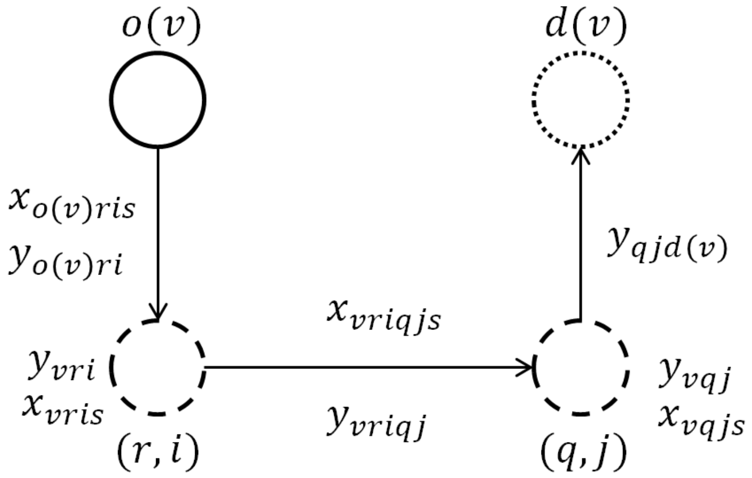

Figure 4.2 – Illustration of nodes and arcs in the model. Each circle is a node, each line is an arc. Dashes are used to illustrate how the origin, voyage and destination nodes are different in nature.

The cargoes loaded on a vessel will be transported along the trade route that is predefined by the shipping company. Each trade route consists of several legs, or sequential sailings, between port visits. Depending on the origin and destination port of the cargo, it will be stored on deck during one or several of these legs. To keep track of cargoes on a vessel, the term *balance categories* is introduced. Let $\mathcal{B}$ be the set of balance categories, indexed by b. Balance categories group volumes of different customers with respect to their starting- and the ending point. For one stretch of a trade route, i.e. when a vessel servicing the trade route Europe-US-East Asia sails from a port in Europe to a port in the US, the vessel may be loaded with balance categories that is (1) shipped from Europe to the US and balance categories that is (2) shipped from Europe to East Asia. In other words, each route can service a predefined subset of balance categories, $\mathcal{B}_r$, where $\mathcal{B}_r \subseteq \mathcal{B}$. Balance categories are illustrated in Figure 4.4.

Unlike container shipping, where containers have more or less standardized dimensions, there are a greater variety in the dimensions of the products transported in the RoRo segment. The cargoes are transported on wheels and can vary from small cars to large break bulk cargo. In the mathematical model this is handled by introducing a set of cargo segments, $\mathcal{P}$. A cargo segment $p \in \mathcal{P}$ may, for example, refer to small cars. The subset $\mathcal{P}_b$ contains the cargo segments associated with balance category b. A product is fully defined through its cargo segment p and balance category b. For each month $m \in \mathcal{M}$, a demand D_{bpm} is to be transported. This demand must either be covered with the company's vessels or by space chartering. The cost of space chartering is C^{SC}_{bpm} per unit. The upper monthly space chartering limit is $\overline{Q}^{SC}_{bpm}$.

Each vessel has a range of decks where the cargo can be stored during transportation. The decks have different capacities and properties related to the dimension and weight of the goods. Let $\mathcal{K}$ be the set of *capacity classes*, or deck types. Each capacity class k points to a certain deck type. Further, the sets $\mathcal{K}_p \subseteq \mathcal{K}$ contain the set of capacity classes that can carry a cargo segment p. The default setup for each vessel on trade route r varies. Let Q_{vrk} be the capacity in each capacity class k when vessel v services trade route r. Figure 4.3 illustrates the connection between decks and capacity classes.

Figure 4.3 – Illustration of capacity classes. Source: WWLBreakbulk (2010)

To keep control of the total volumes of cargo segments loaded on the decks of a vessel on the different stretches of a voyage, a set of *capacity groups* Ξ indexed by ξ is introduced. A capacity group ξ contains the balance categories associated with one stretch of a voyage. Let this set be denoted by $\mathcal{B}_\xi$, and for each capacity group the sum of loads of products (b, p), where $b \in \mathcal{B}_\xi$, should be within the capacity of the vessel. The connection between balance categories and capacity groups is illustrated in Figure 4.4.

To make sure the voyage and maintenance nodes are serviced in time, time variables need to be introduced. Let $t_{o(v)}$ be the starting time for vessel v from its initial position $o(v)$, and t_{ri} be the start time of voyage (r, i). $T^B_{o(v)ris}$ is the time it takes for vessel v to sail ballast from its origin to the start of voyage (r, i) at speed s, and $C^B_{o(v)ris}$ is the

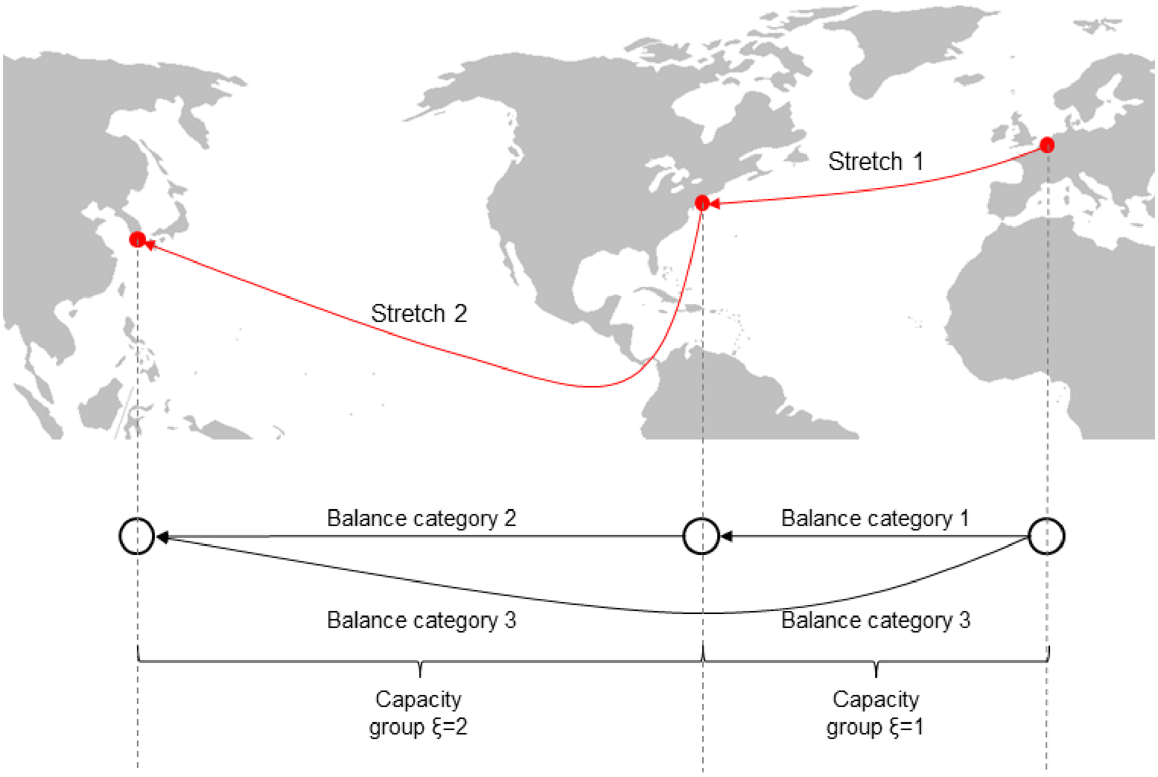

Figure 4.4 – Illustration of two stretches within a voyage and the corresponding balance categories and capacity groups.

corresponding cost. Let T_{vris} denote the time it takes vessel v to sail voyage (r, i) at speed s. The associated cost with this sailing is C_{vris}. Similarly, T^B_{vriqjs} is the time it takes vessel v to sail arc $((r, i), (q, j))$ at speed s with a corresponding cost of C^B_{vriqjs}. Finally, in case of delays, C^P is the penalty cost incurred per day the start of a voyage is delayed.

Let $y_{o(v)ri}$ be 1 if vessel v sails directly from its initial position $o(v)$ to voyage (r, i), and 0 otherwise. $x^B_{o(v)ris}$ is the weight of speed alternative s for vessel v sailing from its initial position $o(v)$ to voyage (r, i). Similarly, let y_{vriqj} be 1 if vessel v sails voyage (r, i) and then sails ballast to voyage (q, j) directly afterwards, and 0 otherwise. x_{vris} is the weight of speed alternative s for vessel v sailing voyage (r, i), and x^B_{vriqjs} is the weight of speed alternative s for the ballast sailing between voyage (r, i) and voyage (q, j) by vessel v. Because the distance from any node to the artificial destination node is zero and there will be no cost related to this sailing, we only need $y_{rid(v)} = 1$ if vessel v sails voyage (r, i) as the last voyage, and 0 otherwise.

Let $y_{o(v)d(v)}$ be 1 if vessel v is not used for any voyages in the planning period and sails directly from its origin to its destination with the corresponding cost $C_{o(v)d(v)}$. Because vessels can be servicing voyages or undergoing maintenance at the start of the planning period, the parameter $E_{o(v)}$ denotes the earliest availability of vessel v. Let E_{ri} and L_{ri} be the earliest and latest allowed time starts of voyage (r, i). Any start after L_{ri} is penalized. Let d_{ri} denote the number of days voyage (r, i) is delayed. The delay tolerance for the model is denoted by the parameter D^{MAX}.

Servicing every contracted voyage in the planning period may in some cases not be possible by only using the set of vessels the company has at its disposal. In order to ensure feasibility, the binary variable y^S_{ri} is introduced. This variable denotes if a contracted voyage is not serviced by the company's fleet. Assigning a value to y^S_{ri} will be penalized with a high cost, C^S, to make the alternative unfavourable. When using this option, it is assumed that the voyage is covered by other transportation options or companies. For these cases, a corresponding capacity for each capacity class is denoted as Q^S_k, which for simplicity is calculated as

$$Q^S_k = \max_{v \in \mathcal{V}, r \in \mathcal{R}} Q_{vrk}, \qquad \forall k \in \mathcal{K}.$$

Finally, the variables l_{ribpk} are introduced. They describe the loaded volumes of product (b, p) on capacity class k on voyage (r, i). The volume of product (b, p) covered with space charter in month m is denoted by z_{bpm}.

4.2 Mathematical Formulation

Objective function

$$\begin{aligned} \min \text{ z} = & \sum_{v \in \mathcal{V}} \sum_{(r,i) \in \mathcal{N}_v} \sum_{s \in \mathcal{S}_v} C_{vris} x_{vris} + \sum_{v \in \mathcal{V}} \sum_{((r,i),(q,j)) \in \mathcal{A}_v} \sum_{s \in \mathcal{S}_v} C^B_{vriqjs} x^B_{vriqjs} + \\ & \sum_{v \in \mathcal{V}} \sum_{(r,i) \in \mathcal{N}_v} \sum_{s \in \mathcal{S}_v} C^B_{o(v)ris} x^B_{o(v)ris} + \sum_{v \in \mathcal{V}} C_{o(v)d(v)} y_{o(v)d(v)} + \\ & \sum_{(r,i) \in \mathcal{N}} C^P d_{ri} + \sum_{b \in \mathcal{B}} \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} C^{SC}_{bpm} z_{bpm} + \sum_{(r,i) \in \mathcal{N}} C^S y^S_{ri} \end{aligned} \tag{4.1}$$

The objective function (4.1) minimizes costs related to operating the fleet of vessels and fulfilling the constraints below. The costs of sailing voyages and ballast between voyages with the speed profile determined by x_{vris} and x^B_{vriqjs} are included in the two first expressions. The third expression addresses costs for the ballast sailing between the origin and the first voyage for each vessel, and the cost of not using a vessel in the planning period is given in the fourth expression. The last line includes the costs for delay in servicing voyage (r, i), costs related to chartering space and the cost of not servicing voyage (r, i) with a vessel from the company's fleet.

Constraints

$$y_{o(v)ri} - \sum_{s \in \mathcal{S}_v} x^B_{o(v)ris} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v, \tag{4.2}$$

$$y_{vri} - \sum_{s \in \mathcal{S}_v} x_{vris} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v, \tag{4.3}$$

$$y_{vriqj} - \sum_{s \in \mathcal{S}_v} x^B_{vriqjs} = 0, \qquad v \in \mathcal{V}, ((r, i), (q, j)) \in \mathcal{A}_v. \tag{4.4}$$

Constraints (4.2)-(4.4) connect the speed and flow variables.

$$\sum_{v \in \mathcal{V}} y_{vri} + y^S_{ri} = 1, \qquad (r, i) \in \mathcal{N}^{\mathcal{C}}. \tag{4.5}$$

Constraints (4.5) ensure that all contracted voyages are serviced.

$$y_{vri} = 1, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}^M_v. \tag{4.6}$$

Constraints (4.6) make sure that vessel v visits a maintenance node if the set $\mathcal{N}^M_v$ is non-empty.

$$\sum_{(r,i) \in \mathcal{N}_v} y_{o(v)ri} = 1 - y_{o(v)d(v)}, \qquad v \in \mathcal{V}, \tag{4.7}$$

$$\sum_{(r,i) \in \mathcal{N}_v} y_{rid(v)} = 1 - y_{o(v)d(v)}, \qquad v \in \mathcal{V}, \tag{4.8}$$

$$y_{vri} - y_{o(v)ri} - \sum_{(q,j) \in \mathcal{N}_v} y_{vqjri} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v, \tag{4.9}$$

$$y_{vri} - y_{rid(v)} - \sum_{(q,j) \in \mathcal{N}_v} y_{vriqj} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v. \tag{4.10}$$

Constraints (4.7)-(4.10) are flow constraints for each vessel.

$$y_{o(v)ri}(t_{o(v)} - t_{ri} + \sum_{s \in \mathcal{S}_v} T^B_{o(v)ris} x^B_{o(v)ris}) \leq 0, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v, \tag{4.11}$$

$$y_{vriqj}(t_{ri} - t_{qj} + \sum_{s \in \mathcal{S}_v} (T^B_{vriqjs} x^B_{vriqjs} + T_{vris} x_{vris})) \leq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v. \tag{4.12}$$

Constraints (4.11) make sure that the start time of the next voyage is greater than or equal to the starting time of the initial position plus the time spent ballast sailing from this position. Constraints (4.12) make sure that the start time of the next voyage is greater than or equal to the start time of the previous voyage plus the time spent sailing it and the ballast sailing between the voyages. In order to implement these constraints in a commercial optimization solver, they need to be linearized. This is shown in Appendix A.2.

$$d_{ri} \geq t_{ri} - L_{ri}, \qquad (r,i) \in \mathcal{N}. \tag{4.13}$$

Constraints (4.13) count the number of days a voyage is delayed.

$$\sum_{r \in \mathcal{R}} \sum_{i \in \mathcal{I}_{rm}} \sum_{k \in \mathcal{K}_p} l_{ribpk} + z_{bpm} = D_{bpm}, \qquad m \in \mathcal{M}, b \in \mathcal{B}, p \in \mathcal{P}_b. \tag{4.14}$$

Constraints (4.14) ensure that the volume transported or space chartered is equal to the monthly demand.

$$z_{bpm} \leq \overline{Q}^{SC}_{bpm}, \qquad b \in \mathcal{B}, p \in \mathcal{P}_b, m \in \mathcal{M}. \tag{4.15}$$

Constraints (4.15) limit the space chartered volume to be lower than or equal to the maximum allowed monthly volume.

$$\sum_{b\in\mathcal{B}_\xi}\sum_{p\in\mathcal{P}_b} l_{ribpk} - \left(\sum_{v\in\mathcal{V}} Q_{vrk}y_{vri} + Q^S_k y^S_{ri}\right) \leq 0, \qquad (r,i)\in\mathcal{N}, k\in\mathcal{K}, \xi\in\Xi. \qquad (4.16)$$

Constraints (4.16) limit the total flow of cargoes in a capacity group on a voyage to be within the capacity of the vessel sailing it.

$$
\begin{aligned}
y_{o(v)ri} &\in \{0,1\}, & v\in\mathcal{V}, (r,i)\in\mathcal{N}_v, && (4.17)\\
y_{vri} &\in \{0,1\}, & v\in\mathcal{V}, (r,i)\in\mathcal{N}_v, && (4.18)\\
y_{vriqj} &\in \{0,1\}, & v\in\mathcal{V}, ((r,i),(q,j))\in\mathcal{A}_v, s\in\mathcal{S}_v, && (4.19)\\
y_{vrid(v)} &\in \{0,1\}, & v\in\mathcal{V}, (r,i)\in\mathcal{N}_v, && (4.20)\\
y_{o(v)d(v)} &\in \{0,1\}, & v\in\mathcal{V}, ((r,i),(q,j))\in\mathcal{A}_v, s\in\mathcal{S}_v, && (4.21)\\
y^S_{ri} &\in \{0,1\}, & (r,i)\in\mathcal{N}_v, && (4.22)\\
x^B_{o(v)ris} &\geq 0, & v\in\mathcal{V}, (r,i)\in\mathcal{N}_v, s\in\mathcal{S}_v, && (4.23)\\
x_{vris} &\geq 0, & v\in\mathcal{V}, ((r,i),(q,j))\in\mathcal{A}_v, s\in\mathcal{S}_v, && (4.24)\\
x^B_{vriqjs} &\geq 0, & v\in\mathcal{V}, ((r,i),(q,j))\in\mathcal{A}_v, s\in\mathcal{S}_v, && (4.25)\\
t_{o(v)} &\geq E_{o(v)}, & v\in\mathcal{V}, && (4.26)\\
t_{ri} &\geq E_{ri}, & (r,i)\in\mathcal{N}, && (4.27)\\
l_{ribpk} &\geq 0, & (r,i)\in\mathcal{N}_v, b\in\mathcal{B}_r, p\in\mathcal{P}_b, k\in\mathcal{K}, && (4.28)\\
z_{bpm} &\geq 0, & b\in\mathcal{B}, p\in\mathcal{P}, m\in\mathcal{M}, && (4.29)\\
0 \leq d_{ri} &\leq D^{MAX}, & (r,i)\in\mathcal{N}. && (4.30)
\end{aligned}
$$

Constraints (4.17)-(4.30) are constraints defining the bounds of the variables.

4.3 Formulation Remarks

The model presented has been developed with the purpose of being as general as possible. Some parts of the model may need to be explained in greater detail.

First, the variables y^S_{ri} are presented as an alternative to the case company's vessels in order to service all voyages in the planning period. In the absence of a spot vessel market, the variables y^S_{ri} can be heavily penalized in the objective function (4.1) with the parameter C^S. In case of a spot market, it is natural to apply (r, i) as indexes to C^S to cope with varying voyage spot charter costs. This means that routing parts of the model can easily be adapted to other liner shipping segments. However, some characteristics of other liner shipping segments, such as cargo transshipment, is omitted from the formulation due to not being common in RoRo shipping.

There are two variables, x_{vris} and x^B_{vriqjs}, that determine the speed when sailing a voyage and sailing ballast to the next voyage respectively. By having separate variables for these two cases, several advantages are obtained. First, it will be possible to exploit different fuel consumption profiles for the two types of sailings. With higher fuel consumption and corresponding cost for sailing the voyage, it may be more cost-efficient to choose a lower speed profile for sailing the voyage and then choose a higher speed profile for the subsequent ballast sailing in order to start the next voyage within its time window. With two separate speed variables, the model is also prepared for an extension where voyage completion deadlines are added. In this case it would be possible to speed up vessels when sailing voyages in order to be in time for the deadline, and then lower the speed when sailing ballast to the next voyage.

Chapter 5

Adaptive Large Neighborhood Search

The framework of large neighborhood search (LNS) heuristics was initially proposed by Shaw (1997), where the heuristic search procedure was tested on vehicle routing problems with promising results. In Section 5.1 we introduce the mechanisms of large neighborhood searches. Then, we show how an LNS may become more efficient by providing it with elements of *adaptability*. These heuristics are called adaptive large neighbourhood searches (ALNS), and was introduced by Ropke and Pisinger (2006). Finally, in Section 5.2 a developed ALNS heuristic for solving the fleet deployment problem is presented.

5.1 The Large Neighborhood Search Mechanisms

The LNS metaheuristic takes in an initial, feasible solution to a problem, and progressively improves the solution by alternately destroying and repairing it through a number of iterations. Unlike other neighborhood search algorithms, where it is normal to perform many small changes to the current solution, the LNS will implement greater changes and be able to explore larger areas of the solution space. By only performing small changes to the solution, such as in local search heuristics, a great number of solutions will be investigated in a short amount of time. However, Ropke and Pisinger (2006) point out the risk of not being able to move from one promising area of the solution space to another.

In the LNS metaheuristic, the neighborhoods are defined by *destroy* and *repair* operators.

The LNS heuristic will typically alternate between an infeasible and a feasible solution. The destroy operation creates the infeasible solution, while the repair operation restores a feasible solution. The design of these operators will determine the performance of the heuristic.

To be able to apply destroy and repair operations, one needs an initial feasible solution to the problem. This can be found in numerous ways. In Korsvik et al. (2011), a tramp ship routing and scheduling problem with split-loads is examined and solved by using a large neighborhood search. Here, the initial solution is obtained by randomly assigning cargoes to feasible vessel schedules. Ropke and Pisinger (2006) suggest that simple construction insertion heuristics may be appropriate. It may also be beneficial to call local search heuristics on the initial solution and/or feasible solutions obtained throughout the search in between the destruction and repair operations, such as in Korsvik et al. (2011). Here, a descent local search consisting of several improvement heuristics is called on the initial solution until a local optimum is reached. This is also done after the repair and destroy sequence is performed in each iteration.

After a solution is repaired, it may be accepted and used in the following iteration, or rejected and the algorithm returns to a previously obtained solution as a basis for future iterations. Several metaheuristic master frameworks may be used for evaluating a solution, such as using tabu criterias or by using *simulated annealing*. The latter was successfully implemented by Ribeiro and Laporte (2012) for evaluating solutions in an LNS for solving a vehicle routing problem. Simulated annealing may help a heuristic escape local optimums. Initially, an inferior move is accepted with a high probability. As the search proceeds, this probability is lowered in each iteration by a *cooling factor*. Towards the end of the search, the system is given time to search nearby neighborhoods for better solutions and only accepts inferior moves with a very low probability.

The Destroy Operators

In each iteration of the search, the destroy operators will remove parts of the solution. Which parts of the solution that are removed is an important decision that has to be made when designing the LNS. With an element of randomness in the destroy operators, there is a higher probability of exploring a different part of the solution in each invocation of the method. Depending on the type of problem and the current solution, different destroy operators could be appropriate. For instance, if only a small part of the solution is destroyed, then the repair heuristics will have difficulties using their full potential to search through the different parts of the solution space. On the other hand, if a very large part of the solution is destroyed, then the repair operation would have to nearly rebuild the solution from scratch and not use the benefits of the heuristic method to the fullest. Finding the right balance of the *degree of destruction* as described above is one of several decisions that must be considered regarding the destroy operators when designing an LNS.

Several destroy operators have been tested and used in the literature. In Korsvik et al. (2011), an operator that removes β random assigned cargoes from the solution is used. In addition to random removals, Pisinger and Ropke (2007), Ribeiro and Laporte (2012), and Ropke and Pisinger (2006) use a worst removal operator and a set of related nodes removal operators for large neighborhood searches on vehicle routing problems. Nodes can be related in several ways, such as in time and distance.

The Repair Operators

After a solution has been destroyed, the objective of the repair operators is to find back to a good, feasible solution. Repair operators can be designed to find an optimal solution given a partial fixed solution, or finding good solutions using heuristics. If the repair operator use an exact solving method, it will take more time to find a solution, but it will be of higher quality. A repair operator based on a heuristic will not necessarily give the same quality as the previous mentioned method, but will be able to resolve the problem much faster.

A few repair operators have commonly been used for solving similar problems. Ropke and Pisinger (2006), Pisinger and Ropke (2007) and Ribeiro and Laporte (2012) use basic greedy insertion and regret-k heuristics. Korsvik et al. (2011) use a constructive insertion heuristic.

Local Search Heuristics

By successively destroying and repairing a solution with various neighborhood operators, the risk of being trapped in a local optimum is significantly reduced in an LNS. Due to their efficiency, local search heuristics have been awarded a great deal of attention as heuristic solution methods in the literature. The nature of large neighborhood searches suggests that local search heuristics can be included and play an important role in the framework, since other operations can help the search escape local optimums. In Korsvik et al. (2011), a descent local search is called in each iteration until a local optimum is found. A set of four local search operators is used in the study: (1) reassign, (2) interchange, (3) 1-split or merge and (4) 2-split or merge.

Adaptive Large Neighborhood Search

When solving problems using an LNS metaheuristic framework, one or several destroy and repair operators can be chosen. A significant benefit of including multiple destroy and repair operators is greater search diversification, making the search able to explore further neighborhoods of the solution. However, some operators may not be as efficient as others, or may be more time-consuming. An *Adaptive Large Neighborhood Search* heuristic extends the LNS framework by allowing multiple destroy and repair operators to be used within the same search, and the frequency of the different destroy and repair operator calls during the search is determined by a weight on each of the operators. The weights will adapt to the instance and be adjusted dynamically throughout the iterations depending on the operator performance (Pisinger and Ropke, 2010). In other words, the heuristic *learns* what works well, and adapts its search to previous successful iterations with certain destroy and repair operators. The adaptive weight adjustment will ensure that operators seldom are used on instances where they are ineffective. Also, a weight adjustment will reflect that some operators may be highly effective in the beginning of the search, but less effective compared to other operators towards the end of the search.

In Pisinger and Ropke (2007), a *roulette wheel selection*-based method for choosing the destroy and repair neighborhood in each iteration is presented. Here, all operators are given a performance score based on past iterations. In the selection process, a higher score implies a higher probability of being chosen. If a past weighted score of an operator i is set as w_i, and there are k operators or neighborhoods, an operator j is chosen with probability

$$\frac{w_j}{\sum\limits_{i=1}^{k} w_i} \tag{5.1}$$

The roulette wheel selection presented by Pisinger and Ropke (2007) selects the repair and destroy operators independently, i.e. two separate selections are performed in each iteration. In some problems, however, certain sets of repair and destroy operators might work well together. Then it may be advisable to rather measure the performance of *pairs* of operators, and subsequently select them in pairs in a roulette wheel selection. In addition to sheer performance, the time consumption of the operators should be considered.

There are also several other advantageous properties associated with the ALNS framework. Pisinger and Ropke (2007) point out that there already exist many well-performing heuristics which can be included as operators in the search. These operators enable the search to structurally explore its neighborhoods. Because of its diversity and capability of dynamically adapting to its own performance, an ALNS will seldom find itself trapped in a local optimum (Pisinger and Ropke, 2007). For the ALNS, opposed to other local search heuristics, instead of choosing from a set of operators one may include all in the search framework, as the search will downgrade not well-working operators. As proposed by Pisinger and Ropke (2007), the more reasonable operators included, the better it performs. Further on, an ALNS may especially work well on tightly constrained problems, where many variables need to be altered in order to find new feasible solutions and also escape local optimums.

5.2 Adaptive Large Neighborhood Search Applied to the Fleet Deployment Problem

In this section we present an ALNS for solving the fleet deployment problem. All decisions made regarding (1) the master level framework, (2) finding an initial solution, (3) selecting destroy operators, (4) selecting repair operators, (5) selecting local search operators and (6) the operator selection procedure are discussed.

Master Level Framework

A pseudo code to illustrate the mechanisms in the developed ALNS is presented in Algorithm 5.1. Here, the lines 7-30 define the iterative search loop. The stopping criteria is based on simulated annealing. In line 8, a destroy and repair operator pair (d, r) from the set of destroy and repair operators $\mathcal{O}$ is selected. This selection is based on weighted scores w_{drj} from the previous segment j of M iterations in the set of segments $\mathcal{J}$. A repair operator r in the set of repair operators $\mathcal{O}^+$ is applied to the destroyed solution $d(y)$ in line 10, where d is a destroy operator in the set of destroy operators $\mathcal{O}^-$ and y is a feasible solution. Also, a randomly selected local search operator l is applied to the solution obtained by $r(d(y))$. In lines 11-21, the new solution is accepted or denied depending on its quality. In line 22, scores based on the quality of y^k is awarded. Then, we check if we have reached a new segment, and if weights should be reset or updated according to Equation (5.8). Finally, the timer t, the temperature T_k^{emp} and the iteration counter k are updated. When the search has been terminated by one of its stopping criteria, the best solution is returned.

To evaluate solutions, we use a methodology where solutions with the fewest unserviced voyages are considered best. Comprehensive testing showed that this procedure gave the most promising results in terms of sheer performance and computational efficiency. When solutions are compared, we check their number of fewest unserviced voyages (see line 11 in Algorithm 5.1). For this evaluation we use the function f, which takes in a solution or a list of solutions and returns the number of unserviced voyages. If two solutions include an equal amount of unserviced voyages, we check if their individual vessel schedules are equal. All solutions with unique vessel schedules and the fewest number of unserviced voyages so far obtained are stored in a list of best solutions ($\overrightarrow{y}^b$). In case a new solution with fewer unserviced voyages is found, the list is reset and only include this new, superior solution. If a solution with a new, unique vessel routing and the same number of unserviced voyages is found, it is added to the list of best solutions. When the search has been terminated, we evaluate all unique solutions in the list and return the one with the lowest operating cost.

On the master level, the acceptance criteria defined by simulated annealing is used in the presented ALNS (see line 17 in Algorithm 5.1). This is chosen because it allows the algorithm to execute inferior moves and thereby may prevent the search from being trapped in a local optimum. With simulated annealing, a solution y^k in an iteration is accepted with probability

$$e^{(-\Delta/T)} \tag{5.2}$$

Here, Δ denotes the difference in the number of unserviced voyages between the solution y^k obtained in the current iteration and solutions in the list of best solutions $\overrightarrow{y}^b$. In mathematical terms, $\Delta = f(y^k) - f(\overrightarrow{y}^b)$. However, in the presented algorithm, a relative

Algorithm 5.1. Pseudo Code for Adaptive Large Neighborhood Search

Input: an initial, feasible solution y;
$\overrightarrow{y}^b = (y^b) = (y); w_{dr,0} = \frac{1}{|\mathcal{O}^-||\mathcal{O}^+|}$;
Set max iterations = K, start counter $k = 0$;
Set max time = T^{MAX} seconds, start counter $t = 0$;
Set segment size = M, set reset parameter = M^0, start segment counter $j = 0$;
Start simulated annealing, $T^{emp}_k = T^{emp}_{start}$, $c =$ cooling factor;
while ($k < K$ and $t < T^{MAX}$) **do**
Select destroy and repair pair $(d, r) \in \mathcal{O}$ using w_{drj} according to Equation (5.7);
Randomly select local search operator l;
$y^k = l(r(d(y)))$;
if $f(y^k) < f(\overrightarrow{y}^b)$ **then**
$\overrightarrow{y}^b = (y^k)$;
$y = y^k$;
else if $f(y^k) == f(\overrightarrow{y}^b)$ and y^k is a new schedule **then**
Add y^k to list of best solutions $\overrightarrow{y}^b$;
$y = y^k$;
else if accept (y^k, y) according to probability given by Equation (5.2) **then**
$y = y^k$;
else
Return to a random solution in $\overrightarrow{y}^b$;
end if
Award scores $\sigma_1, \sigma_2, \sigma_3$;
if (k % $M == 0$) **then**
Update segment $j = j + 1$;
if (k % $M^0 == 0$) **then**
Reset weights $w_{drj} = \frac{1}{|\mathcal{O}^-||\mathcal{O}^+|}$;
else
Update weights w_{drj} according to Equation (5.8);
end if
end if
update t, $T^{emp}_{k+1} = c \cdot T^{emp}_k$;
$k = k + 1$;
end while
return Best solution in $\overrightarrow{y}^b$;

difference is chosen implemented, i.e. Δ is divided by the number of unserviced voyages in the current best solutions. The temperature T_k^{emp} of the search in iteration k starts at T_{start}^{emp} and is decreased in every iteration according to a *cooling rate* $0 < c < 1$. The new temperature in iteration $k + 1$ is calculated as $T_{k+1}^{emp} = c \cdot T_k^{emp}$. Improving solutions or solutions with new, unique vessel schedules are always accepted.

There are a few things one must consider when setting T_{start}^{emp} and c. A too high start temperature will ensure early diversification, but may seize unnecessary amounts of time. The cooling factor must not cool down the solution too slow, as inferior moves will be accepted too often and not give the search enough time to work with good solutions. On the other hand, it must not cool down the solution too fast, which increases the risk of being trapped in a local optimum.

Finding an Initial Solution

An initial, feasible solution for the presented ALNS is found by using a regret-k repair operator as a construction heuristic. This was successfully implemented by Andersen (2010) in an ALNS for solving a network transition problem in liner shipping. In contrast to the results obtained by Andersen (2010), where $k = 2$ performed best, initial testing showed that a degree of $k = 4$ gave the best results for the problem studied in this thesis. The regret-k repair operator is also used in the repair phase of the search and is presented in further detail below.

Selecting Destroy Operators

The developed ALNS for the FDP uses five different destroy operators. These operators take in a solution where all voyages are assigned, and returns an infeasible solution where some voyages are marked as unassigned. To limit the number of removal requests in the largest test instances, an upper removal limit, γ, is imposed on each destroy operator. This parameter denotes the maximum size of the list of unassigned voyages. If γ is too low, Ropke and Pisinger (2006) point out that the heuristic will not be able to move much around in each iteration, and may become trapped in a local optimum despite acceptance frameworks such as simulated annealing. On the other hand, a too high level of destruction is undesirable, as it will require the repair operator to rebuild large parts of the solution. Ropke and Pisinger (2006) show that insertion heuristics perform poorly for this purpose. Similarly, let α be a parameter denoting the maximum number of voyages which is allowed removed as a percentage of the total number of voyages. The random removal destroy operator uses α as the probability of removing an assigned voyage from the current solution when looping through all assignments.

Trade Route Removal

The Trade Route Removal operator removes voyages, either unserviced or serviced by vessels, belonging to a specific trade route r randomly selected from the set of trade routes $\mathcal{R}$. This operator is motivated by the possibility the repair operators now have to reallocate voyages related in terms of geography.

Month Removal

Similar to Trade Route Removal, the Month Removal operator removes all assigned or unserviced voyages with their target start date in a specific month m or in its following month $m+1$, where m is randomly selected from the set of months $\mathcal{M} \setminus \{|M|\}$. The idea behind this operator is that the voyages removed are related through their placement in time. Figures 5.1 and 5.2 show how a solution is affected by applying the Month Removal destroy operator.

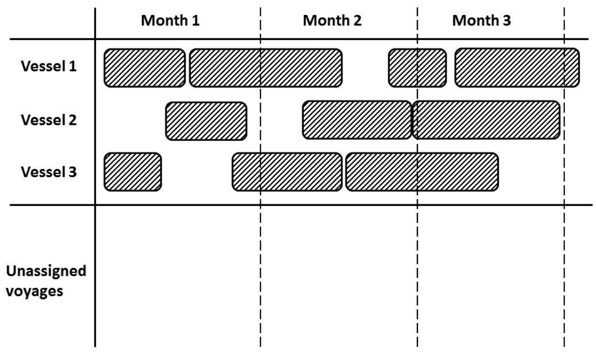

Figure 5.1 – Feasible solution before Month Removal

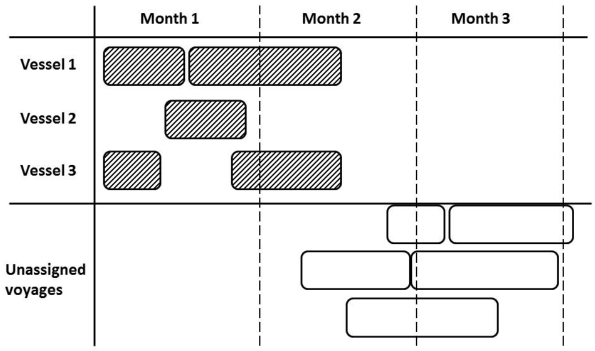

Figure 5.2 – Solution after Month Removal

Vessels Removal

The Vessels Removal operator removes all assigned voyages belonging to three vessels v_1, v_2 and v_3 randomly selected from the set of vessels $\mathcal{V}$. To be as general as possible, restrictions have been made in case there are fewer than three vessels. By leaving three vessels with no schedule, any unassigned voyages may be reassigned in a more cost-efficient way. A pseudo code for the Vessels Removal operator is presented in Algorithm 5.2.

Algorithm 5.2. Pseudo Code for Vessels Removal Destroy Operator

Function DestroyThreeVessels($y \in$\{solutions\}, $v \in \mathcal{V}$, $(r,i) \in \mathcal{N}$);
vessel: $v =$ a randomly selected vessel from $\mathcal{V}$
set of vessels randomly selected: $\mathcal{V}^U \subseteq \mathcal{V}$
set of unassigned voyages: $\mathcal{N}^U \subset \mathcal{N}$
set of voyages not serviced: $\mathcal{N}^S \subseteq \mathcal{N}$
for all $(v \in \mathcal{V}^U)$ **do**
    Remove all voyages (r,i) assigned to v and add them to $\mathcal{N}^U$;
end for
while $(|\mathcal{N}^U| < \gamma)$ and $(|\mathcal{N}^U| < \alpha \cdot |\mathcal{N}|)$ and $(|\mathcal{N}^S| > 0)$ **do**
    Remove voyages $(r,i) \in \mathcal{N}^S$ and add them to $\mathcal{N}^U$;
    Update $|\mathcal{N}^U|$ and $|\mathcal{N}^S|$;
end while
return destroyed solution $d(y)$;

Random Removal

The Random Removal operator is the simplest of the five proposed destroy operation heuristics, and removes a random set of assigned voyages while iterating through the solution. A random assigned voyage is removed with probability α up to the maximum total amount γ. This operator is important because it adds diversification in the search process.

Port Removal

The Port Removal operator removes all inbound and outbound voyages of a randomly selected port over a given time period from a solution. Just as the Trade Route Removal operator, the motivation behind this operator is the possibility to swap voyages related in geography and time.

Selecting Repair Operators

A few different well-working repair operators from the literature have been reviewed previously, and are used as inspiration when determining what might work when applied to the FDP. In this section we present four repair operators that repair destroyed solutions in different ways.

Greedy Insertion Heuristic

One of the repair operators used is a basic insertion heuristic. The heuristic iterates over all unassigned voyages and attempts to place them in open time windows in the preliminary vessel schedules. For each unassigned voyage (r, i), all vessels $v \in \mathcal{V}$ are examined according to their current voyage assignments. The unassigned voyage is assigned to the vessel where the cheapest insertion possibility is found. For simplicity, all insertion possibilities are evaluated given that all vessels sail at their given maximum service speeds. All unassigned voyages are shuffled before the heuristic is initiated to ensure diversification.

Mathematically, let ΔZ_{vri} denote the cost added to the objective function Z of inserting voyage (r, i) from the set of unassigned voyages $\mathcal{N}^U$ at the cheapest possible insertion in the schedule of vessel v. If no possible insertion is found in the schedule of vessel v, then set $\Delta Z_{vri} = \infty$. After looping through all vessels, the following calculation is executed:

$$(v) := \underset{v \in \mathcal{V}}{\operatorname{argmin}} \, \Delta Z_{vri} \qquad \forall (r, i) \in \mathcal{N}^U. \tag{5.3}$$

Following this calculation, voyage (r, i) is assigned to vessel v where the cheapest possible insertion is found, and the voyage start time t_{ri} is determined by its own time window and by the time window available at vessel v. Assigning this start time is important for the remaining iterations of unassigned voyages. Lastly, if no feasible insertion is found, the voyage is set as unserviced. A pseudo code for the Greedy Insertion Heuristic is presented in Algorithm 5.3.

Algorithm 5.3. Pseudo Code for Greedy Insertion Heuristic

Function GreedyInsertion($d(y)$)
set of unassigned voyages: $\mathcal{N}^U = \mathcal{N} \setminus d(y)$;
for all ($(r, i) \in \mathcal{N}^U$) **do**
    least costly assignment $\Delta Z_{vri} = \infty, \forall v \in \mathcal{V}$;
    for all ($v \in \mathcal{V}$) **do**
        if cost of possible insertion $< \Delta Z_{vri}$ **then**
            $v' = v$;
            $\Delta Z_{v'ri}$ = cost of insertion;
        end if
    end for
    if $\Delta Z_{vri} < \infty$ **then**
        insert (r, i) according to Equation (5.3);
    else
        assign (r, i) to y^S_{ri};
    end if
end for
clear list of unassigned voyages N^U;
return repaired solution $r(d(y))$

Deep Greedy Insertion Heuristic

Similarly to the Greedy Insertion Heuristic, the Deep Greedy Insertion Heuristic iterates over all unassigned voyages and attempts to place them in open time windows in the vessel schedules. However, while the former places the unassigned voyages on the go, the Deep Greedy Insertion Heuristic evaluates the cost of all possible insertions, i.e. it loops over all unassigned voyages and vessel schedules before making any insertion decisions. After an unserviced voyage has been assigned to a vessel, all remaining unassigned voyages are iterated again to find the second cheapest insertion. Despite being significantly more computationally challenging, better decisions are likely to be made as the search is deeper. If the iteration procedure reveals that an unassigned voyage has no possible insertions, it is set as unserviced.

For the Deep Greedy Insertion Heuristic, let ΔZ_{vri} denote the cost added to the objective function Z of inserting voyage (r, i) from the set of unassigned voyages $\mathcal{N}^U$ at the cheapest possibility in the schedule of vessel v. If no insertion possibility is found in vessel v, then set $\Delta Z_{vri} = \infty$. After looping through all vessels and unassigned voyages, the following calculation is executed:

$$(v, r, i) := \underset{v \in \mathcal{V}, (r,i) \in \mathcal{N}^U}{\operatorname{argmin}} \Delta Z_{vri} \tag{5.4}$$

Regret-k heuristics

The Greedy Insertion and Deep Greedy Insertion heuristics calculate and insert the cheapest requests first, which means that difficult requests are postponed until the end of the iterations. However, at this point, most of the schedules have started to fill up and possible voyage-vessel combinations are unlikely. Regret heuristics attempt to solve this problem by considering the effect of *not* inserting a request at the cheapest possible placement.

Let ΔZ_{ri}^k denote the change in the objective value by inserting the unassigned voyage (r, i) at its kth cheapest possible insertion in the vessel schedules. This means that, e.g., ΔZ_{ri}^2 refers to the second cheapest possible insertion of voyage (r, i). If no possible insertions are found, let $\Delta Z_{ri}^k = \infty$. In this example, $k = 2$, and the request chosen to be inserted by the regret heuristic is decided according to

$$(r, i) := \underset{(r,i) \in \mathcal{N}^U}{\operatorname{argmax}} \left(\Delta Z_{ri}^2 - \Delta Z_{ri}^1 \right). \tag{5.5}$$

In Equation (5.5), the difference between inserting the unassigned voyage at the second cheapest and the cheapest possible placement in the vessel schedules are calculated, and the voyage with the greatest difference is chosen to be inserted. In other words, the heuristic evaluates the consequence of not inserting a voyage at its cheapest possible insertion. For some voyages, which are flexible and have many possible insertions, the difference may be small. For other voyages, there may only be one single possible insertion in the current set of vessel schedules, making the difference large.

The heuristic can be extended to be as general as possible and consider the k cheapest possible insertions of voyage (r, i), as shown by Equation (5.6):

$$(r, i) := \underset{(r,i) \in \mathcal{N}^U}{\operatorname{argmax}} \left(\sum_{h=2}^{k} \Delta Z_{ri}^h - \Delta Z_{ri}^1 \right). \tag{5.6}$$

Compared to a regret heuristic with $k = 2$, regret-k heuristics are more likely able to detect when the possibilities of inserting an unassigned voyage becomes limited in the future and may make more considered decisions. On the other hand, including too much information may cause the heuristic to perform too conservative insertions and the optimal insertion may be overlooked.

Partial Fixed MIP Repairing

Here, a destroyed solution is sent to the mathematical program defined in Section 4.1 as a partial fixed MIP problem. The variables fixed are y_{vri} and y_{ri}^{S}, i.e. the decisions regarding which voyage to be sailed by which vessel or marked as unserviced. The unassigned voyages remain unfixed and are optimized in the solver. Although more time-consuming than the heuristic methods presented earlier, this repair operator attempts to find the optimum solution given any fixed variables. It is important to ensure that enough variables are fixed in the solution sent to the solver, so that the optimization process does not become too slow and fails to find good solutions within a time limit T^{MIP}. The parameters α and γ help limit the degree of destruction.

Local Search Heuristics

Inspired by Korsvik et al. (2011), the ALNS was extended with a few local search operators which where found effective in initial testing. Several others were also tested, but these were found computationally challenging and ineffective on larger and realistic instances.

Reassign

The Reassign local search operator attempts to reallocate any unserviced voyages to vessel schedules, no matter the cost of these individual insertions. This operator can also adjust vessel sailing times and the start times of assigned voyages, in order to service unserviced voyages.

1-Interchange

The 1-Interchange local search operator iterates all vessel schedules and unserviced voyages, and attempts to reallocate a serviced voyage from one vessel to another, i.e. an interchange. An interchange is performed if it makes room for an unserviced voyage in the new vessel schedules.

Destroy and Repair Operator Selection

An alternation between the different destroy and repair operators may provide the search with a wide specter of neighborhoods to explore. For selecting an operator pair in each iteration, a roulette wheel selection such as defined by Equation (5.1) suggested by Pisinger and Ropke (2007) is used. Equation (5.7) below is a revised version of Equation (5.1) with the needed notation to formulate the selection of destroy and repair operators in pairs and separate weights in segments of iterations. Let $w_{d'r'j}$ be the weight of operator pair (d', r') in segment $j \in \mathcal{J}$. The probability of selecting this operator pair during the iterations in segment j is calculated as

$$\frac{w_{d'r'j}}{\sum\limits_{(d,r)\in\mathcal{O}} w_{drj}} \qquad \forall j \in \mathcal{J}. \tag{5.7}$$

How the weights w_{drj} are assigned to the different pairs of repair and destroy operators is presented in the following section.

Adaptive Weight Adjustment

The score of an operator pair is recorded when a solution is evaluated, and represents a measure of how well the heuristic performed in this iteration. Consequently, a high score means that the heuristic has been performing successfully. To give the different operators sufficient opportunities to "prove their worth", the search is divided into segments of iterations. A segment corresponds to a number of iterations M in the ALNS heuristic loop. In the reviewed literature, the suggested segment size M is between 50 and 100 iterations. For every M iterations, the scores are reset and the weights w_{drj} are updated. The new weights are updated according to the scores in the past iterations as well as the weights used in the past segment. Let w_{drj} denote the weight of heuristic (d, r) used in segment j. When deciding the weights to be used in iterations belonging to segment $j+1$, the following formula presented by Pisinger and Ropke (2007) is used:

$$w_{dr,(j+1)} = w_{drj}(1-\rho) + \rho \frac{\overline{\pi}_{drj}}{a_{drj}} \tag{5.8}$$

Here, $\overline{\pi}_{drj}$ is the normalized score and a_{drj} is the number of calls of heuristics (d, r) in the past segment j. The *reaction factor* ρ is introduced to adjust how rapid the weights change from segment to segment according to their effectiveness. The score $\overline{\pi}_{drj}$ of operator pairs in a segment is incremented by the *score adjustment parameters* σ_1, σ_2, σ_3 or zero in an iteration, depending on the recorded performance. Before awarded to an operator pair, the score is normalized with respect to the computational time consumed by the operator pair. The score parameters are explained more in detail in Table 5.1. Their values are given so that $\sigma_1 > \sigma_2 > \sigma_3 > 0$. The final weights $w_{dr,(j+1)}$ obtained are used to select operator pairs in each iteration in the next segment according to Equation (5.7). For every M^0 iterations, the weight values are reset to the weights initially defined by $w_{dr,0}$. This is done to ensure that operators which may perform ineffective in early parts of the search does not have an unreasonable small probability of being selected later on, when they may perform better.

Table 5.1 – Score adjustment parameters

Score parameter	**Description**
σ_1	The iteration resulted in a new global best solution where fewer voyages are unserviced.
σ_2	The iteration resulted in a new, unique vessel schedule with the same number of unserviced voyages, i.e. the vessel routing is altered.
σ_3	The iteration resulted in a worse solution, but it was accepted as a part of simulated annealing.
0	Neither of the above.

Figure 5.3 presents an example of how weights develop during a search. The weight reset mechanism can be observed every 20th segment. Because of simulated annealing and because it is harder to find good moves towards higher iteration numbers, the given scores and thereby the weights are evenly decreasing throughout the search.

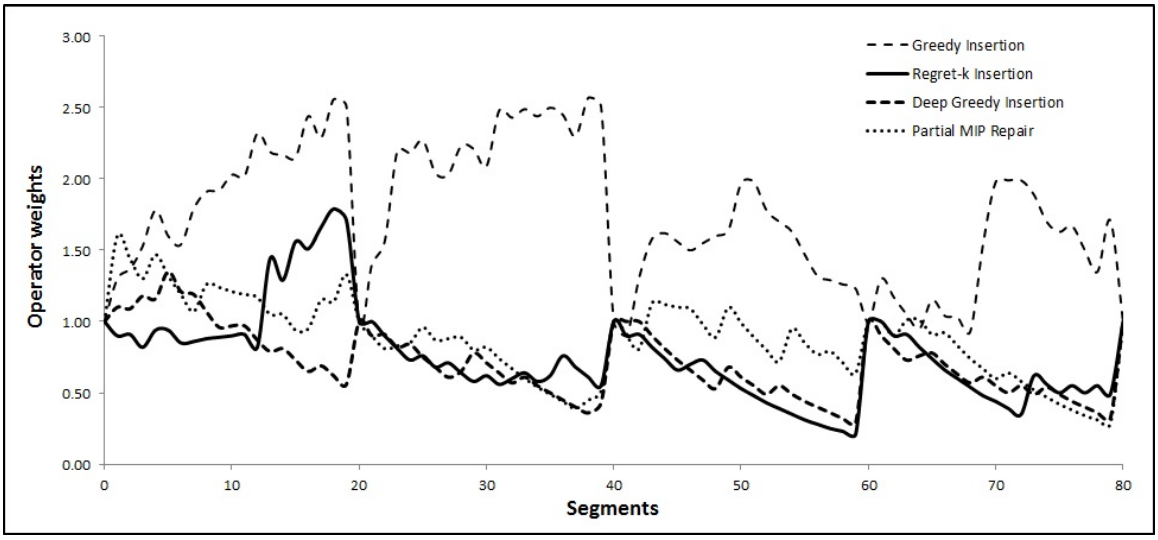

Figure 5.3 – Illustration of how weights of operators develop during a search. The weights in the example are from an execution of the ALNS heuristic on S24_V222_T9_M2. The weight developments shown are from combinations of the Trade Route removal heuristic and all the repair operators. Here, each segment consists of 50 iterations.

Chapter 6

Rolling Horizon Heuristics

An increasingly popular approach for solving hard planning problems with long planning horizons is rolling horizon heuristics (RHH). As stated in Section 1.4, this heuristic approach was successfully implemented by Andersson et al. (2014) in order to solve the FDP over the desired 6-10 months planning period. Due to the problem similarities, it is reasonable to believe that a rolling horizon heuristic has the potential of being an excellent solving method for the problem studied in this thesis.

In Section 6.1, an overview and the general mechanisms of an RHH is presented. Then, a description of how we applied the heuristic to solve the FDP is proposed in Section 6.2.

6.1 The Rolling Horizon Heuristic Mechanisms

The main idea behind rolling horizon heuristics is to solve the problem over a *rolling horizon*. This means to solve parts of the problem iteratively over the whole planning period. For each iteration, a sub-problem for each sub-horizon is solved. The sub-horizon includes a primary period and a forecasting period. When solving the problem in each sub-horizon, the properties of the original problem are kept intact in the primary period, while properties in the forecasting period are somehow simplified. This simplification can, for example, be a relaxation of integer variables. The decisions made in the primary period are partly or fully fixed in the next iteration. This iterative process continues until decisions for the whole planning horizon have been made. The rolling horizon approach divides the problem into smaller parts which makes it easier to solve.

The method of rolling horizon heuristics has long been in use for solving problems within production planning. In Baker (1977), the effectiveness of rolling schedules for production planning is investigated. The authors present findings that demonstrate how rolling schedules produce good results, usually within 10 % of optimality. A general framework for analyzing rolling schedules were proposed by Baker and Peterson (1979), with particular focus on the effects of factors such as the length of the planning horizon. The authors examine a cost model for production planning, and discover that performance improves monotonically as the planning horizon is extended. Stauffer and Liebling (1997) present a rolling horizon algorithm for an aluminum production plant. Here, the job portfolio for the plant is only certain for a limited period, and future orders are continuously received. A tabu search-based heuristic solves the sub-problem in each iteration. Mercé and Fontan (2003) use a MIP-based algorithm with a rolling horizon framework to solve the multi-item capacitated lotsizing problem. For each iteration, a size-reduced MIP for the current sub-horizon is solved. In the paper, Mercé and Fontan (2003) suggest two different strategies when it comes to fixing variables in each sub-horizon. The first strategy is to fix all decisions made in the previous primary section, while the second strategy is to fix only production decisions. Other variables, such as production quantities, may be adjusted later in the algorithm. Mercé and Fontan (2003) found the best results by using the second strategy. Dellaert and Jeunet (2003) address the multi-level lotsizing problem, and show that the performance of fixed-horizon methods can be greatly improved in a rolling horizon environment. Bredström et al. (2013) demonstrate a method to solve LP problems with uncertain right-hand sides by applying it to planning problems where rolling horizons are used. In order to simplify and reduce the computational load of the forecasting section in each sub-horizon, Bredström et al. (2013) successfully relaxes binary variables in the forecasting section.

In addition to Andersson et al. (2014), there exist a few other publications were RHH have been used for maritime planning problems. Sherali et al. (1999) propose a MIP model for routing and scheduling ships in a maritime transportation system. For the specific problem studied in the paper, a typical planning horizon is 300 days, and the related scenario consists of 25,920,029 variables and 5,058,360 constraints in total. This complex problem were successfully solved by using a rolling horizon approach. Bredström and Rönnqvist (2006) solve a combined supply chain and ship routing problem over a 40-day period by using a heuristic method based on a rolling time horizon. In their heuristic, Bredström and Rönnqvist (2006) relax binary variables denoting combinations of vessels and routes in the forecasting section of each sub-horizon. Computational experiments from applying the heuristic on real world instances suggest that the solution method in many cases can be very efficient. Finally, Rakke et al. (2011) present an RHH for a large scale ship routing and inventory management problem. Here, the authors suggest that binary variables in the forecasting section should be made continuous, and that the length of the forecasting section should be twice as long as the primary section. The computational study shows that the algorithm returns good solutions to real world data instances within a relatively short amount of time.

It is natural to elaborate on the RHH proposed in Andersson et al. (2014), considering the similarity to our problem. In the study, many of the techniques previously presented are used, such as (1) relaxing the binary variables as a simplification strategy in the forecasting section, and (2) fixing binary variables related to servicing voyages. Time and vessel speed variables may be adjusted later in the algorithm. Andersson et al. (2014) show that the RHH obtained good solutions within reasonable times and performed much better for all realistic instances than a MIP model.

The Rolling Horizon Algorithm

Figure 6.1 illustrates the mechanism of a rolling horizon heuristic, and is inspired by similar figures in Mercé and Fontan (2003), Rakke et al. (2011), and Andersson et al. (2014). Each sub-horizon consists of a primary period and a forecasting period. For each iteration, the current sub-problem for the corresponding sub-horizon is solved. Let us denote K as the number of iterations to be performed. At iteration k, the sub-problem for the sub-horizon H_k is solved. Further more, $S_P(H_k)$ and $S_F(H_k)$ describes the primary period S_P and forecasting period S_F in sub-horizon H_k, respectively. When the computation in each iteration has been done, the solutions in $S_P(H_k)$ are fixed according to the predefined fixing strategy, and the primary and forecasting periods are updated to $S_P(H_{k+1})$ and $S_F(H_{k+1})$. Also, the new sub-horizon is chosen by moving forward a given number of time units. Now, the sub-problem H_{k+1} is solved. In the last period of the horizon, no forecasting section is used, and all decisions made are final.

6.2 Rolling Horizon Heuristics Applied to the Fleet Deployment Problem

Algorithm 6.1 shows the pseudo code for the implemented RHH. The primary period $S_P(H_k)$ contains the same properties and constraints as the exact problem. Determining

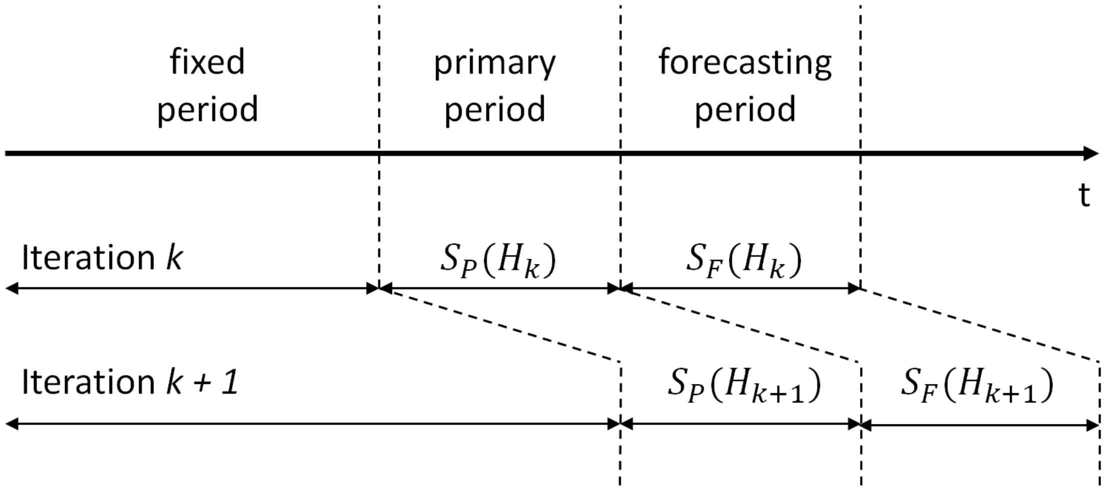

Figure 6.1 – The rolling horizon heuristics mechanism.

the length of the primary period is a decision that must be made when designing the RHH. Andersson et al. (2014) address that a trade-off between solution quality and computation time has to be made when deciding the length of the primary period, as it is reasonable to assume that a longer primary period will provide better solutions. A primary period of one month was chosen in the presented RHH. Decisions regarding the variables $y_{o(v)ri}$, y_{vri} and y_{vriqj} are fixed after each primary period has been solved. This means that variables related to speed, load, start times, a vessel's final destination and unserviced voyages may be adjusted later on.

Algorithm 6.1. Pseudo Code for Rolling Horizon Heuristic

```
Input: length of primary periods T_{S_P}; length of forecasting periods T_{S_F}; number of sub-horizons K;
set k = 1;
while k < K do
    define H_k from T_{S_P} and T_{S_F};
    solve the model for S_P(H_k) and S_F(H_k);
    fix flow variables y_{o(v)ri}, y_{vri} and y_{vriqj} in S_P(H_k);
    k = k + 1;
end while
solve the model for the final sub-horizon S_P(H_K);
return solution;
```

In order to take future events into consideration, a forecasting period $S_F(H_k)$ is included in each sub-horizon. This will affect the decisions made in the primary period, and likely construct a more holistic solution. Baker and Peterson (1979) propose that longer forecasting periods will provide better solutions, but at the cost of longer computing times. Therefore, we face a trade-off between length and solution quality. When testing the effect of different lengths of primary and forecasting periods in Andersson et al. (2014), it was pointed out that the increased quality of the solution was not significant, and with a much higher solution time. This was also tested in Fischer, Nokhart, and Olsen (2014), where it was concluded that the RHH with a forecasting period of two months produced equal or better solutions within a much shorter amount of time, compared to the RHH with a forecasting period set as the rest of the planning period. Based on these findings, we have decided to set the length of the forecasting period to two months. The sub-horizon in the developed RHH thereby consists of a primary period of one month, and a forecasting period of two months ahead.

For the forecasting period, a simplification strategy can be made. Several publications have successfully relaxed binary variables (e.g. Bredström and Rönnqvist (2006), Andersson et al. (2014)), which lowers the computational load of solving the problem. However, less information is used than in the original formulation. Again, we have a trade-off between maintaining information and keeping the problem smaller and less complex. The presented RHH relaxes all binary variables related to routing in the forecasting period.

Chapter 7

Computational Study I

In this chapter the implementation and a computational study of the presented models are described and discussed. The MIP, ALNS and RHH models were coded in Java and run in the Eclipse Luna integrated development environment for Java Developers Release 1 (4.4.1) programming interface. The MIP, the fixed LP-problem related to the Partial Fixed MIP repair operator and the sub-problem in each sub-horizon in the RHH were solved by using Gurobi Optimizer version 6.0 (Gurobi Optimization, 2014) in the Java interface. All of the models were run on a cluster of computers with HP DL165 G6, 2 x AMD Opteron 2431 2.4 GHz, 24 GB of RAM, and 164 GB SAS 15000rpm.

In Section 7.1 we provide a description of the various problem instances used in this computational study. Then, the computational performances of the MIP, ALNS and RHH are presented and thoroughly discussed in Sections 7.2, 7.3 and 7.4, respectively. Finally, a comparison of the different solving methods is presented in Section 7.5. Here we also examine the effect of different time window widths and separate speed variables.

7.1 Description of Problem Instances

The data used for the models are provided by the case company in Microsoft Excel format, and is collected directly by the Java models from the spreadsheets. However, values for some of the parameters in the model are not included in the provided data. First, the cost of not servicing a voyage is not given. As explained in Section 4.1, this cost must be set very high to heavily penalize this option. If the cost of not servicing a voyage is not sufficiently high, we risk that this option is cheaper than using the case company's vessels to cover the voyage. However, setting the cost too high will increase the numerical objective value significantly. In solutions where unserviced voyages must be present to ensure feasibility, the resulting high objective value will overshadow the operating costs. Initial testing of different costs for not servicing voyages showed that the quality of solutions increased with a lower cost, but it also increased the solution time of the models. The main objective of the models are to provide useful and quality solutions to the case company. Reducing the solution time is only useful if the outcome of the models is of high quality. Based on these considerations, the cost of not servicing voyages, C^S, was set to the value that provided the highest solution quality, \$100 millions.

In the mathematical formulation, vessels can sail directly from their origin to their artificial destination, i.e. they may not be used in the planning period. Not using a vessel could simply be considered as a fixed cost for the company, but also as an opportunity to charter out the vessel to other companies and provide an income. However, we have not included this opportunity and the cost of not using vessels during the planning period, $C_{o(v)d(v)}$, is therefore set to zero.

As presented in Section 4.1, the tolerance for starting a voyage after the time window is denoted with the parameter D^{MAX} and any delay is penalized with the cost C^P. The possibility of starting a voyage later than the time window will give more flexibility to the model, as discussed in Section 4.3. However, for this part of the thesis we choose to leave the delay feature out by setting the parameter D^{MAX} to zero. The delay penalty cost is thereby not relevant, but a value of \$200,000 for the parameter C^P is suggested by the case company. A summary of the parameters discussed and the corresponding values can be seen in Table 7.1.

Table 7.1 – Summary of assumed parameter values

Parameter	**Description**	**Value**
C^S	Cost of not servicing voyages	1.00E+8
$C_{o(v)d(v)}$	Cost of not using vessels	0
D^{MAX}	Days of tolerated delay	0
C^P	Daily delay penalty cost	\$200,000

Problem instances and modifications

The test data consists of five different problem instances from the case company, with different size and complexity. The simplest and smallest problem instance contains two vessels and five voyages to be serviced over a planning period of three months, while the largest and most complex problem instance has 24 vessels, 222 voyages and a planning period of nine months. A summary of the characteristics of the different problem instances is presented in Table 7.2. The problem instances are named so that, e.g., an instance called "S5_V52_T7" has five ships, 52 voyages and a planning period of seven months. In addition to the number of vessels, voyages and length of the planning period, the width of the time windows for the voyages determine the problem complexity.

Table 7.2 – Problem instances characteristics

Instance	Vessels	Voyages	Months	Speed profiles	Cargo segments
S2_V5_T3	2	5	3	1	1
S5_V52_T7	5	52	7	3	4
S5_V77_T8	5	77	8	2	4
S16_V109_T9	16	109	9	3	4
S24_V222_T9	24	222	9	3	4

None of the presented problem instances contain any maintenance nodes, meaning that all vessels are available for servicing voyages during the whole planning period. As can be seen in Table 7.2, the problem instances also have different characteristics for the number of speed profiles available and cargo segments considered. Additional speed profiles and cargo segments increase the problem size and complexity even further.

As is discussed in Section 7.5, the width of the time windows has a great impact on the complexity of the problems. Extending the time period when a voyage can start increases the number of possible combinations each vessel can sail, thus increasing the solution space. To be able to discuss the impact of the time windows' width, new modified problem instances are generated. Each voyage has one target day that will remain the same, while the earliest and latest start dates on each side of the target day is modified as described below. The width of the time windows are set depending on when a voyage is scheduled to start. It is assumed that the voyages in the first months have less flexibility than the voyages in the later months of the planning period. Therefore, the first months consistently have tighter time windows in all modifications.

Modification 1 (M1) For all the months in the planning period, the earliest and latest start of the voyages are the target day $\pm$ 1 day, resulting in a time window of three days.

Modification 2 (M2) For the two first months in the planning period, the earliest and latest start dates of the voyages are the target day ± 1 day, resulting in a time window of three days. For the rest of the months in the planning period, the earliest and latest start dates of the voyages are the target day ± 2 days, resulting in a time window of five days.

Modification 3 (M3) For the two first months in the planning period, the earliest and latest start dates of the voyages are the target day ± 1 day, resulting in a time window of three days. For the two subsequent months the earliest and latest start dates of the voyages are the target day ± 2 days, resulting in a time window of five days. For the rest of the planning period the earliest and latest start dates of the voyages are the target day ± 3 days, resulting in a time window of seven days.

Depending on the modification, M1, M2 or M3 is added to the problem instances' label. The complete label for a problem instance is then $SX^S\_VX^V\_TX^T\_MX^M$. Here, X^S is the number of ships, X^V is the number of voyages, X^T is the number of periods, and X^M is the modification ID.

With all modifications and the original versions, there are in total 20 problem instances with different characteristics. However, some of the original problem instances have time window widths of over 60 days. In this thesis we decide to only focus on the more realistic time windows resulting from the modifications M1, M2 and M3. There are no preferences for when a voyage should be started within its time window, meaning that all of the days in the time window are equally valid start times.

7.2 Mixed-Integer Program - Implementation and Results

In this section we first describe how the mathematical formulation given by (4.1)-(4.30) was implemented as a MIP in commercial optimization software and the considerations that followed in the process. Then, the results from solving the FDPs defined by the problem instances are presented and discussed.

Tuning the model

During the implementation of the mathematical model in Java, several decisions were made when tuning the model in order to make it solve efficiently. When solving mixed-integer linear programming problems, the Gurobi Solver generally uses a LP-based branch-and-bound algorithm. To get what Lundgren et al. (2010) describe as a better balance in the search tree, it will be branched on the sum of variables. Prioritizing variables that have high objective function coefficients or high impact on the objective function will provoke a better shape of the search tree and result in a more efficient search. In the mathematical model there are especially two variables that have great impact on the objective function: (1) y^S_{ri} denoting if voyage (r, i) is not serviced, and (2) y_{vri} denoting which vessel that will visit voyage (r, i). The binary constraints for these variables are prioritized in anticipation of a more efficient search tree.

The high complexity of some problem instances require the MIP model to run for a long period of time in order to find optimal solutions. A maximum running time of 10,000 seconds is set for the MIP-model. If optimality is not found within this time, the best feasible solution is reported, together with the gap from the bound.

Results and discussion

The results from solving the problem instances with the MIP model using Gurobi are presented in Table 7.3. The operating costs reported are the objective value minus the costs of any unserviced voyages in the solution. The gap presented in the table is defined as

$$Gap(\%) = \frac{\text{Best integer solution} - \text{Best bound}}{\text{Best bound}}$$

As can be seen from Table 7.3, the MIP model is able to find near optimal solutions in four of the five problem instances for all modifications. For the first two modifications of problem instance S16_V109_T9, it can be seen that the model is still running after 10,000 seconds while their best solutions have a gap that are below 0.1 and 1.0 %, respectively. In these cases, the model is able to find good solutions but solutions that are not proven to be optimal, which explains why the model continues until the termination time of 10,000 seconds. In Section 7.5 this observation is further elaborated and the solution times for when good-quality solutions are reported.

For the largest problem instance, S24_V222_T9, the solver was not able to find any optimal solutions within the available time for neither of the modifications. With the increased solution space that arises with the wider time windows, it is clear that the problem

Table 7.3 – Results from solving problem instances with MIP

Instance	OC[1]	Gap (%)	UV[2]	ST[3]
S2_V5_T3_M1	1.05500E+5	0.00	0	0.14
S5_V52_T7_M1	3.15082E+8	0.00	0	17
S5_V77_T8_M1	1.52057E+8	0.00	3	29
S16_V109_T9_M1	2.57669E+8	0.07	4	10,000
S24_V222_T9_M1	2.07788E+8	101	15	10,000
S2_V5_T3_M2	1.05500E+5	0.00	0	0.01
S5_V52_T7_M2	3.17940E+8	0.00	0	352
S5_V77_T8_M2	1.51577E+8	0.38	3	10,000
S16_V109_T9_M2	2.51344E+8	0.88	3	10,000
S24_V222_T9_M2	1.65835E+8	940	61	10,000
S2_V5_T3_M3	1.05500E+5	0.00	0	0.01
S5_V52_T7_M3	3.17737E+8	0.01	0	1,145
S5_V77_T8_M3	1.51464E+8	0.67	3	10,000
S16_V109_T9_M3	2.51776E+8	1.9	3	10,000
S24_V222_T9_M3	1.43754E+8	1,413	89	10,000

[1] OC = Operating cost

[2] UV = Number of unserviced voyages

[3] ST = Solution time in seconds

becomes more complex to solve. This can especially be seen by observing the number of unserviced voyages for the largest problem instance. We conclude that the presented MIP model is not able to solve problems of this complexity within the available time of 10,000 seconds.

For the problem instances S5_V52_T7_M1 and S5_V77_T8_M1, the program is able to return the optimal solution within a short amount of time. However, for modifications 2 and 3, where the solution space is increased, also the smaller problem instances require significant more time to be solved. These results show that the solution time for the MIP model increases significantly with complexity of problem instances. It also indicates that heuristic approaches are needed to solve realistic fleet deployment planning problems within a reasonable amount of time.

The vessel schedules obtained when solving problem instance S2_V5_T3_M1 is presented in Figure 7.1.

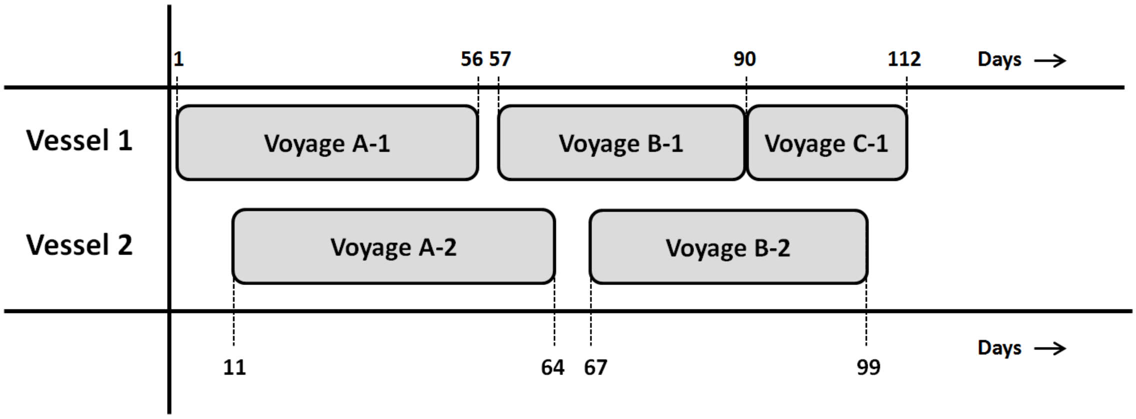

Figure 7.1 – Schedule for problem instance S2_V5_T3_M1. The empty slots between voyage sailings are time allocated to either sailing ballast or waiting for the time window of the next voyage to start.

7.3 Adaptive Large Neighborhood Search - Implementation and Results

In this section we present the results obtained by solving the problem instances with the developed ALNS. First, a description of the determined parameter values are given. Then, the results are presented and discussed.

Parameter settings

There are many parameters associated with the ALNS which need to be determined prior to applying the heuristic to the FDP, as presented in Section 5.2. Regarding the removal heuristics, an upper limit of γ element removals must be imposed. Important circumstances around this parameter is discussed in depth in Section 5.2. The values of γ is set according to suggested values in the reviewed literature (Ropke and Pisinger, 2006, Ribeiro and Laporte, 2012), and was thereafter slightly modified given the results of initial testing. In addition to γ, the parameter α denotes the upper limit on removal requests as a percentage of the total number of voyages. This parameter is also set according to values in the reviewed literature. In addition to being an upper limit, α is used by the random removal destroy operator as the probability of removing an assigned voyage from the current solution.

An issue encountered during the initial testing was whether to automatically treat unserviced voyages as unassigned requests or not in each iteration. On one hand, these assignments are very unfavorable and should be included in a vessel schedule as quickly as possible. Therefore, the repair operators should be given the opportunity to try and find possible insertions for these. On the other hand, removing all unserviced voyages from solutions may initially result in many unassigned requests (especially for the larger problem instances), which greatly increases the computational time of the repair operators. Therefore, an upper limit of $\gamma/2$ was imposed on the destruction of unserviced voyages before further requests specified by the current removal operator were removed. The mechanisms of the local search heuristics may help eliminate the majority of unserviced voyages promptly.

The Partial Fixed MIP repair operator presented in Section 5.2 sends a destroyed solution to a solver as a partially fixed linear problem. To give this repair operator enough time to find optimal or near-optimal solutions, but not become too time consuming, an upper time limit T^{MIP} is imposed on the program, and as soon as the gap is below 1.00 %, the solution is accepted.

As presented in Section 5.2, the scores σ_1, σ_2, σ_3 and 0 are used to adjust the weights w_{drj} after every M iterations. In Equation (5.8), $\overline{\pi}_{drj}$ is the sum of the scores given to destroy-repair operator pair $(d, r) \in \mathcal{O}$ obtained through the M iterations in segment j, normalized with respect to time consumption, ρ is the reaction factor to previous weights, and a_{drj} is the number of calls of the destroy-repair operators in segment j. The initial values of the weights, $w_{dr,0}$, are equal and sum up to 1. Every M^0 iterations, the weights are reset to the values given initially by $w_{dr,0}$.

$$w_{dr,(j+1)} = w_{drj}(1-\rho) + \rho \frac{\overline{\pi}_{drj}}{a_{drj}} \tag{5.8 revisited}$$

At the master level search framework, simulated annealing is used to control the acceptance criteria. Here, the start temperature T^{emp}_{start} of the solution and the cooling factor c through the search are parameter values that must be specified. Figure 7.2 is produced from the output of an initial test run of the ALNS on S16_V109_T9_M3, and shows how the number of voyages not serviced in the current solution and in the best solution are functions of the current iteration. The characteristics of Figure 7.2 illustrates the mechanisms in a simulated annealing framework. In the beginning of the search, inferior moves are accepted more often and the graph is clearly volatile. However, as the temperature cools down, only less dramatic inferior moves are accepted, and such moves are accepted less frequently.

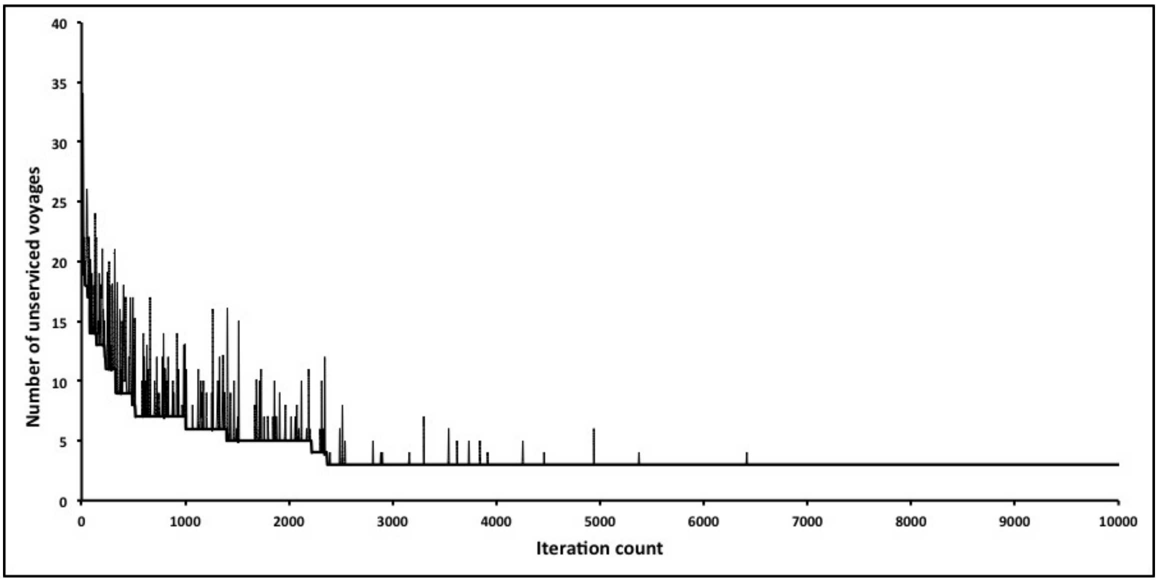

Figure 7.2 – Number of unserviced voyages as a function of the iteration count. The upper graph shows accepted solutions with more unserviced voyages as a part of simulated annealing, while the lower bound represents the currently best known solution with fewest unserviced voyages. The graph is produced from the data of every fifth iteration from the output of an initial test run of the ALNS on problem instance S16_V109_T9_M3.

Table 7.4 shows the final parameter values used when solving the problem instances with the ALNS in this computational study. All selected parameter values are a combination of the suggested values in Ribeiro and Laporte (2012) and Ropke and Pisinger (2006), and problem-specific experience obtained through the initial testing process of the algorithm.

Results and discussion

The results from solving all problem instances with the ALNS are summarized in Table 7.5. All of the problem instances were solved five times and were terminated either by the time limit of 10,000 seconds or by reaching the maximum count of 25,000 iterations. Columns 2-4 report the best solution found for each problem instance throughout the runs. The solution time denotes the search time elapsed before the best reported solution was found. In columns 5-7, the average of the obtained solutions are presented. Similarly, the solution time denotes the average time elapsed in each run before the best reported solution was found. Finally, the operating cost is the objective value minus the cost of unserviced voyages for both the best solution and the average solution.

Table 7.5 shows that the presented ALNS is capable of finding good solutions for most of the problem instances within reasonable computation times. For all instances but

Table 7.4 – Values of ALNS parameters

Parameter	Description	Value
γ	Upper limit of assigned voyages removed in each iteration.	30
T^{MIP}	The maximum time the Partial Fixed MIP repair operator is allowed to work when called, in seconds.	30
α	Upper limit of assigned voyages removed in each iteration, as a percentage of number of voyages.	30 %
σ_1	Score given when last removal heuristic resulted in a new schedule where fewer voyages are not serviced.	35
σ_2	Score given when last removal heuristic resulted in a new schedule with the same number of unserviced voyages, i.e. different vessel routing.	15
σ_3	Score given when last removal heuristic resulted in a worse solution, but it was accepted as a part of simulated annealing.	5
M	The size of each search segment, in number of iterations.	50
M^0	Number of iterations before weights are reset.	1000
ρ	The reaction factor used in calculating operator weights.	0.1
$w_{dr,0}$	Initial destroy-repair operator pair weights. Here, $\|O\|$ is the number of destroy and repair operator pairs.	$1/(\|O\|)$
T^{emp}_{start}	The start temperature in simulated annealing.	1.000
c	The cooling factor in simulated annealing.	0.99975

the two largest, the search finds optimal or near-optimal solutions. The solution for S16_V109_T9_M1 is not good, but for the two last modifications the quality of the solutions is significantly better with gaps below 3 %. This effect may be caused by the difference in time windows, which is further elaborated in Section 7.5. For the largest instance, S24_V222_T9, better solutions are frequently found during the search until the time limit is reached. This means that the provided time is not sufficient for the ALNS to find good solutions to the largest problem instances.

Observations made by monitoring the search indicate that the time consumption related to the repair operation in each iteration grows significantly with the problem size. Figure 7.3 shows the number of iterations per time unit for all instances except S2_V5_T3, and illustrates how each iteration becomes very time-consuming for larger instances. For example, for instances S24_V222_T9, the search performs down to only 0.3 iterations per second, while for instances S16_V109_T9, the search performs 1.1-1.5 iterations per second.

Table 7.5 – The results from solving the problem instances with the ALNS.

Instance	Best Solutions				Average Solutions		
	OC[1]	UV[2]	Gap(%)	ST[3]	Gap(%)	UV[2]	ST[3]
S2_V5_T3_M1	1.05500E+5	0	0.00	0.00	0.00	0	0.00
S5_V52_T7_M1	3.15143E+8	0	0.02	3.9	0.08	0	1,403
S5_V77_T8_M1	1.52057E+8	3	0.00	117	0.06	3	918
S16_V109_T9_M1	2.45389E+8	5	13	4,650	17	5	4,356
S24_V222_T9_M1	2.02422E+8	18	136	9,760	166	21	6,967
S2_V5_T3_M2	1.05500E+5	0	0.00	0.00	0.09	0	0.00
S5_V52_T7_M2	3.18721E+8	0	0.25	16	0.35	0	16
S5_V77_T8_M2	1.52541E+8	3	0.60	64	0.77	3	81
S16_V109_T9_M2	2.58493E+8	3	2.2	8,097	9.4	3	4,740
S24_V222_T9_M2	2.08937E+8	15	184	5,292	200	16	6,582
S2_V5_T3_M3	1.05500E+5	0	0.00	0.00	0.00	0	0.00
S5_V52_T7_M3	3.18367E+8	0	0.21	18	0.46	0	37
S5_V77_T8_M3	1.52502E+8	3	0.90	130	1.2	3	178
S16_V109_T9_M3	2.56661E+8	3	2.8	2,156	3.2	3	4,597
S24_V222_T9_M3	2.20193E+9	11	121	9,703	157	13	8,351

[1] OC = Operating cost

[2] UV = Number of unserviced voyages

[3] ST = Solution time in seconds

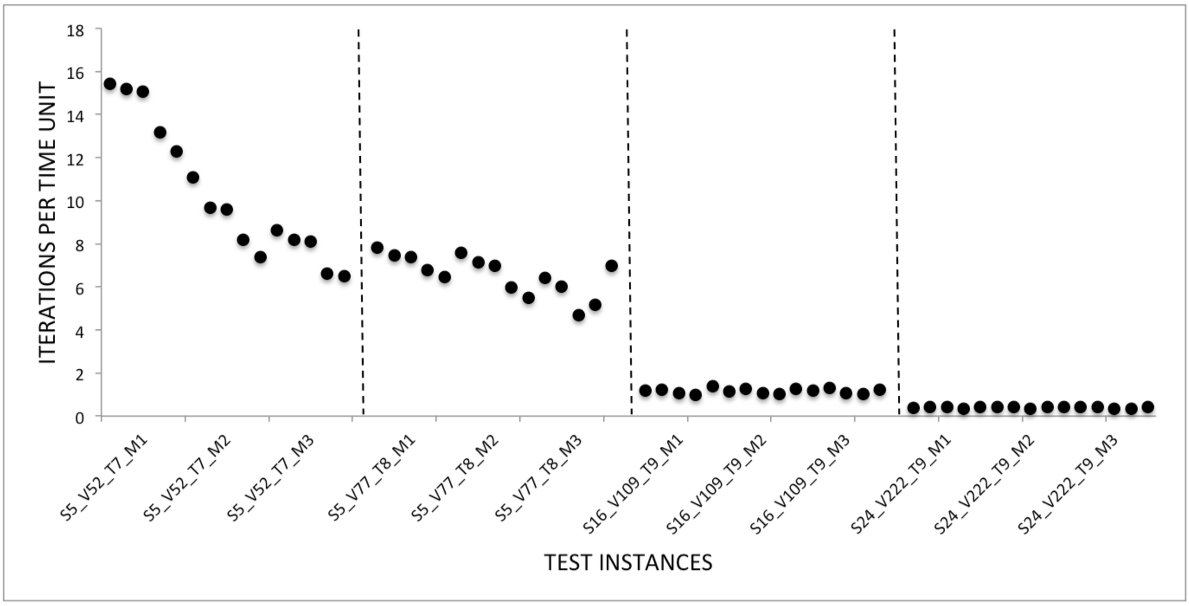

Figure 7.3 – Number of iterations per time unit for various problem instances.

7.4 Rolling Horizon Heuristic - Implementation and Results

In this section, the implementation of and the results obtained by applying the RHH to the FDP are presented. First we describe the tuning of the model, before discussing the results obtained by solving the problem instances with the RHH.

Tuning of the model

When implementing the RHH model in Java, some considerations were made to make the model run as intended. To be able to compare the performance of the RHH with the MIP model, the maximum running time should be set to the same as for the MIP model. However, setting a total time limit for the model is not sufficient, as we have to ensure that every sub-problem is allocated time to find a solution. Therefore, each sub-problem is limited to 10,000 seconds divided by the number of sub-horizons. The number of sub-horizons equals the number of months in the planning period. This means that for the largest problem instances with a planning period of nine months, the available time for each iteration becomes 10,000/9 = 1,111 seconds.

The RHH is based on dividing the planning period into different sub-horizons, and where each voyage will be assigned in one, and only one, sub-horizon. This was done in the implementation by only considering voyages with the latest start time, L_{ri}, included in the current sub-horizon. We also use the latest start time for a voyage to connect variables with the respective bounds in the sub-horizon. The variable y_{vri}, denoting if vessel v sails voyage (r, i), will be relaxed to be continuous if the latest start time of voyage (r, i) is in the forecasting part of the sub-horizon, and it will likewise have binary conditions if the the latest start time is in the primary period of the current sub-horizon. Using latest start times as reference times ensure that no variables are fixed until all voyages that could possibly be connected are considered. This means that the variable y_{vriqj}, denoting if vessel v sails voyage (r, i) and then ballast to (q, j), will only be fixed when a voyage (q, j) has its latest start time in the currently considered primary period.

Results and discussion

The results for the RHH are presented in Table 7.6. The operating cost reported is the objective value deducted the cost of any unserviced voyages. The gap is calculated in the same way as for the MIP, i.e. as the objective value minus the best bound divided by the best bound in percentage value. The best bound is obtained from the MIP.

Table 7.6 – Results from solving the problem instances with the RHH

Instance	OC[1]	Gap (%)	UV[2]	ST[3]
S2_V5_T3_M1	1.05500E+5	0.00	0	0.01
S5_V52_T7_M1	3.15149E+8	0.02	0	6.8
S5_V77_T8_M1	1.52502E+8	0.10	3	15
S16_V109_T9_M1	2.62067E+8	0.74	4	66
S24_V222_T9_M1	2.16095E+8	20	8	1,828
S2_V5_T3_M2	1.05500E+5	0.00	0	0.00
S5_V52_T7_M2	3.18215E+8	0.09	0	36
S5_V77_T8_M2	1.51637E+8	45	5	75
S16_V109_T9_M2	2.55004E+8	1.6	3	121
S24_V222_T9_M2	2.28334E+8	4.3	4	3,420
S2_V5_T3_M3	1.05500E+5	0.00	0	0.00
S5_V52_T7_M3	3.17889E+8	0.06	0	40
S5_V77_T8_M3	1.51695E+8	0.72	3	86
S16_V109_T9_M3	2.53730E+8	2.3	3	210
S24_V222_T9_M3	2.26286E+8	4.8	4	5,520

[1] OC = Operating cost
[2] UV = Unserviced voyages
[3] ST = Solution time in seconds

The results in Table 7.6 show that the RHH is capable of solving all problem instances within a reasonable amount of time. Only for solving the largest problem instances, the method requires a significant amount of time. Despite the trade-off between quality and solution time in heuristics pointed out in the relevant literature, the results show an excellent solution quality for most of the problem instances. The solutions with a gap below 1 % are considered as very good, while the solutions with a gap below 5 % are considered as acceptable. Here, 9 out of 15 of the solutions are good, and 13 out of 15 qualify as acceptable solutions.

It was only two of the problem instances that reported a gap above 5 %. This could either be because the MIP model was not able to report a best bound close to the optimal solution, or that the solutions obtained by the RHH is of poor quality. The latter may have occurred due to several reasons. For the first modification of the largest problem instance, the model was not able to service all voyages with the company's fleet, which is penalized and results in a high objective value. As can be seen for the modifications with broader time windows for the same problem instance, fewer unserviced voyages are reported. The same occurs for problem instance S5_V77_T8_M2, where the solution obtained includes more unserviced voyages than in the two other modifications. A reasonable explanation for the high gap in both cases is that the current modification causes voyages to have their upper time window limit, L_{ri}, in the first day of a month. This could complicate the fixing of the variables when the month is shifted from being a forecasting period to becoming a primary period. If the LP-relaxed binary variable that determines to sail the voyage, y_{vri}, has a value close to zero, it may not be feasible according to the fixed plan from the previous period to change the value to 1. Two suggestions to prevent this problem are discussed in Chapter 8.

Aside from these two particular cases, the results clearly show that the RHH is performing very well by offering quality solutions within a short amount of time.

7.5 Comparing the Solving Methods and General Remarks

In this section, the presented MIP, ALNS and RHH results are compared and briefly discussed. The time each solving method needs to find quality solutions, the effect of speed optimization and the effect of wider time windows are also analyzed.

The main results of the three different solving methods are summarized in Table 7.7. For the smaller problem instances, up to S5_V77_T8, the MIP model is able to prove optimality faster than or near equally fast as the other solving methods. When the complexity increases, as for modifications 2 and 3 and the larger problem instances, the RHH is the superior solving method, and is capable of finding optimal or near-optimal solutions within the time limits specified. The average gap column in Table 7.7 illustrates this. The RHH is best or equally best for all but three problem instances, and it is also the fastest solving method in all these cases.

The ALNS appears less effective than the RHH. Even though it is capable of finding optimal solutions to nine of the fifteen problem instances, it is only the fastest solving method for two of them. However, these two are smaller instances where the solution times for the different solving methods are fairly similar. For the larger instances it does

Table 7.7 – MIP, ALNS & RHH results

	MIP			ALNS			RHH		
Instance	**UV**[1]	**Gap(%)**	**ST**[2]	**UV**[1]	**Gap(%)**	**ST**[2]	**UV**[1]	**Gap(%)**	**ST**[2]
S2_V5_T3_M1	0	0.00	0.14	0	0.00	0.00	0	0.00	0.01
S5_V52_T7_M1	0	0.00	17	0	0.02	3.9	0	0.02	6.8
S5_V77_T8_M1	3	0.00	29	3	0.00	117	3	0.10	15
S16_V109_T9_M1	4	0.07	10,000	5	13	4,650	4	0.74	66
S24_V222_T9_M1	15	101	10,000	18	136	9,760	8	20	1,828
S2_V5_T3_M2	0	0.00	0.01	0	0.00	0.00	0	0.00	0.00
S5_V52_T7_M2	0	0.00	352	0	0.25	16	0	0.09	36
S5_V77_T8_M2	3	0.38	10,000	3	0.60	64	5	45	75
S16_V109_T9_M2	3	0.88	10,000	3	2.2	8,097	3	1.55	121
S24_V222_T9_M2	61	940	10,000	15	184	5,292	4	4.27	3,420
S2_V5_T3_M3	0	0.00	0.01	0	0.00	0.00	0	0.00	0.00
S5_V52_T7_M3	0	0.01	1,145	0	0.21	18	0	0.06	40
S5_V77_T8_M3	3	0.67	10,000	3	0.90	130	3	0.72	86
S16_V109_T9_M3	3	1.9	10,000	3	2.8	2,156	3	2.3	210
S24_V222_T9_M3	89	1,413	10,000	11	121	9,703	4	4.8	5,520

[1] UV = Number of unserviced voyages
[2] ST = Solution time in seconds

find better solutions than the MIP, but it is still inferior to the solutions provided by the RHH.

Overall, the RHH appears as an excellent method for solving the FDP. Despite its simplicity, the method is capable of finding optimal or near-optimal solutions within the given time limits for large problem instances where the MIP does not provide sufficient results. When planning the fleet deployment for longer horizons, the RHH may serve as a great decision-support tool for shipping companies, especially if a good solution is quickly required. This could occur in several real life situations, e.g., when plans may need to be adjusted, optional voyages are on the table and should be accepted or rejected, or for budgeting purposes.

The time until quality and good solutions are found

As stated in Nygreen et al. (1998), three important keywords in an efficient operations research modeling process are (1) reality representation, (2) ease of communication and (3) solution speed. The classic trade-off is presented as reality representation and solution speed, where an increase of the reality representation will usually lead to a decrease in solution speed. The results presented in the previous sections are based on methods that aim to solve a very realistic representation of the problem, and finding optimal solutions. However, as previously stated, the MIP did in some cases find good solutions with a low gap, but because they are not proven optimal, the model will not terminate until the time limit of 10,000 seconds is reached. This motivated us to examine when the models find solutions that are of high quality. When setting the acceptance criteria for the high quality solutions, the high cost of not servicing voyages in the objective value must be taken into account. For the three largest problem instances, the best bound from the MIP includes several voyages that could not be serviced, which results in a high numeric value of the best bound. In previous literature high quality solutions are assumed to be solutions with objective values below a given threshold, e.g. 1 %, of the best bound. However, due to the high cost of unserviced voyages, a low threshold value does not guarantee good operating costs. For the problem instances with unserviced voyages in their best bound, the numeric objective value accepted will be very high, with high corresponding operating costs. Because of the high cost of unserviced voyages, we were able to identify the number of unserviced voyages in the best bound and subtract this cost in order to find the best bound for the operating cost. To find when the solutions have an operating cost with high quality, we calculate new termination gaps for the problem instances based on Equation (7.1). These are gaps for the objective value that corresponds to operating costs with a gap below 1 % and 5 % respectively.

$$Gap^T = \frac{(1 + Gap^S)(BB^{MIP} - UV^{BB} \cdot C^S) - BB^{MIP}}{BB^{MIP}} \tag{7.1}$$

Gap^T is the termination gap for the objective value, Gap^S is the accepted gap for the operating costs, BB^{MIP} is the best bound found when solving MIP to optimality, UV^{BB} is the number of unserviced voyages in the best bound and C^S is the cost of unserviced voyages. The new termination gaps corresponding to the quality schedule solutions are presented in column 2 and 5 in Table 7.8, together with the time each solving method needed to find a solution within this gap.

Table 7.8 – The time in seconds until good and quality solutions are found. The gaps in columns 2 and 6, which are calculated with Equation (7.1), are used as objective value termination criteria that corresponds to a 1 % and 5 % gap of the operating cost of the schedule, respectively. A dash means that no solution satisfying the termination criterion was found within 10,000 seconds.

	Termination gap of OC[1] = 1 %				**Termination gap of OC[1] = 5 %**			
Instance	**Gap (%)**	**MIP**	**ALNS**	**RHH**	**Gap (%)**	**MIP**	**ALNS**	**RHH**
S2_V5_T3_M1	1.00	0.14	0.01	0.01	5.00	0.14	0.01	0.01
S5_V52_T7_M1	1.00	4.2	6.6	6.8	5.00	2.1	6.6	6.8
S5_V77_T8_M1	0.34	25	-	15	1.68	8.2	-	15
S16_V109_T9_M1	0.39	3,569	-	-	1.96	203	-	66
S24_V222_T9_M1	0.29	-	-	-	1.46	-	-	-
S2_V5_T3_M2	1.00	0.01	0.00	0.00	5.00	0.01	0.00	0.00
S5_V52_T7_M2	1.00	215	15	36	5.00	64	15	36
S5_V77_T8_M2	0.33	-	-	-	1.67	139	-	-
S16_V109_T9_M2	0.45	-	-	-	2.26	4,951	-	121
S24_V222_T9_M2	0.37	-	-	-	1.68	-	-	-
S2_V5_T3_M3	1.00	0.01	0.00	0.00	5.00	0.01	0.00	0.00
S5_V52_T7_M3	1.00	228	16	40	5.00	48	16	40
S5_V77_T8_M3	0.33	-	-	-	1.63	120	-	86
S16_V109_T9_M3	0.45	-	-	-	2.23	7,056	-	-
S24_V222_T9_M3	0.33	-	-	-	1.66	-	-	-

[1] OC = Operating costs

For the RHH, setting a termination gap will terminate every sub-problem at the given gap. This means that the model could accept a solution with a 1 % or 5 % gap in the early sub-problems, and it may not be possible to continue the solution process and retain the low gap. However, as was showed when solving to optimality in Section 7.4, the RHH model is able to find good solutions within a reasonable amount of time. Therefore, the RHH results when solving to optimality are reported in Table 7.8 and are compared with the time the MIP and ALNS need to find good solutions.

The results in Table 7.8 show several interesting findings. First, as was suspected from the analysis of the original MIP results, the MIP is able to find a good solution for problem instance S16_V109_T9 in a much shorter amount of time than when solving to optimality. With the first modification it is able to find a solution with an operating cost gap of 5 % and 1 % within 203 and 3,569 seconds, respectively. Also, for the two other modifications of problem instance S16_V109_T9, the MIP model is able to find good solutions with an operating cost gap below 5 % within a much shorter amount of time than the 10,000 seconds available.

Second, the results in Table 7.8 show that the ALNS model is only capable of finding good solutions for the two smallest problem instances. The RHH is able to find more good solutions than the ALNS, but it is actually the MIP that manages to find the largest number of good solutions. However, when the MIP is the only model that manages to find good solutions, it requires 3,569 and 7,056 seconds for problem instance S16_V109_T9_M1 and S24_V222_T9_M3, respectively. This is considered as a significant amount of time and the solution speed is thus not sufficiently low for the MIP to be considered as an efficient operations research modeling process. In comparison, for problem instance S16_V109_T9_M2 the RHH is able to find a good solution with a gap for the operating cost below 5 % in only 121 seconds compared to the MIP's 4,951 seconds. This again illustrates that the RHH is superior when it comes to solution speed and quality.

The effect of speed optimization

One of the properties that distinguishes the work presented in this thesis from most relevant previous work is the inclusion of speed optimization. By adding the variables x_{vris} and x^B_{vriqjs}, the model will also find the optimal speed profile for sailing voyage and sailing ballast in between voyages. In this section the effect of speed optimization is discussed, as well as the extension made in the presented formulation that enables speed optimization for the ballast sailing separately.

Sequential routing and speed optimization

As described in Section 4.1, speed variables are included in the formulation to make the model as realistic as possible. However, an increase in the number of variables increases the problem complexity. Motivated by the work done in Norstad et al. (2011), we look at how integrated speed optimization increases the solution time of the model. We modify the model to sequentially solve the routing part of the FDP with only the highest speed profile available, and afterwards we perform the speed optimization with all speed profiles included. A flowchart of these processes is presented in Figure 7.4.

To illustrate how the sequential routing and speed optimization performs, the least complex modification of the problem instances was chosen and the results are reported in Table 7.9. In columns 4-6 the operating cost, number of unserviced voyages and the solution time of solving the routing problem with the highest speed profile are reported. In the last three columns, we report the same results when speed-optimizing the route obtained with all speed profiles available. For comparison reasons the MIP results are also presented. Problem instance S2_V5_T3 only contains one speed profile, and is therefore not included in these results.

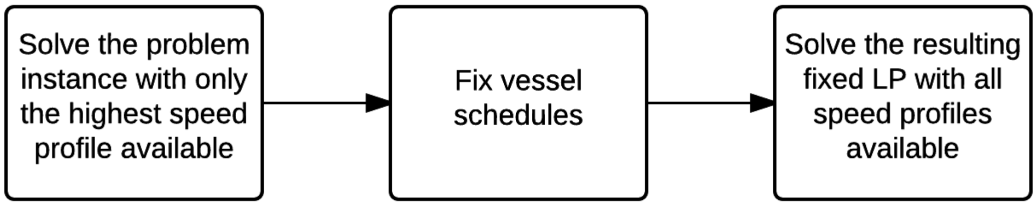

Figure 7.4 – Flowchart of how problem instances are sequentially solved with only the highest speed profile available before speed optimizing the resulting deployment plan

Table 7.9 – Comparison of original MIP with sequential routing and speed optimization. The gap reported for the routing solution and the solution from routing and speed optimization is based on the best bound from the original MIP when solving to optimality. Since the speed optimization is performed on the schedules already found, the number of unserviced voyages will be the same in the solution for routing and speed optimization and for routing with highest speed profile.

	Routing with Highest Speed Profile			**Routing and Speed Optimization**		**MIP results**		
Instance	**Gap(%)**	**UV[1]**	**ST[2]**	**Gap(%)**	**ST[2]**	**Gap(%)**	**UV[1]**	**ST[2]**
S5_V52_T7_M1	1.4	0	1.6	0.01	1.7	0.00	0	17
S5_V77_T8_M1	1.1	3	6.3	0.03	6.5	0.00	3	29
S16_V109_T9_M1	3.1	4	234	0.41	235	0.07	4	10,000
S24_V222_T9_M1	12	7	4,201	8.6	4,204	101	15	10,000

[1] UV = Unserviced voyages
[2] ST = Solution time in seconds

The results from the routing part of the solving method, which are reported in column 2-4 in Table 7.9, show that good vessel routing solutions are found within a significant lower solution time compared to the original MIP results. When speed optimizing the routes, the model is able to further improve the quality of the solutions within a short amount of time. The results indicate that by integrating speed optimization in the FDP, the complexity of the model increases and a longer solution time is experienced. This can especially be seen for the largest problem instance, where the original MIP model is not able to find good solutions within the time available. By sequentially routing and speed optimizing the same problem instance, a better solution is found within only 4,204 seconds.

The great impact on the solution time of the largest problem instances shows how speed optimization adds complexity to the problem. When comparing the final solution time in Table 7.9 it shows that the speed optimization only requires a maximum of 3 seconds for the largest problem instance in order to speed optimize the routing. The routing part of the problem requires significant more time than the speed optimization part and is for the largest problem instance considered as too long to be used as en efficient OR-model. Based on these findings, we developed a sequential routing and speed optimization model using the RHH for the routing part and then speed optimize the routing by using the MIP model. Only the highest speed profile was used when routing, and the results are compared with the results from Table 7.9 and the results from the original RHH.

Table 7.10 – Sequential routing with highest speed profiles using RHH and speed optimizing with MIP compared with sequential routing with highest speed profile and speed optimization using MIP and original RHH results

	Routing using RHH and speed optimization using MIP			**Routing and speed optimization using MIP**			**RHH results**		
Instance	**Gap(%)**	**UV**[1]	**ST$_{RASO}$**[2]	**Gap(%)**	**UV**[1]	**ST$_{RASO}$**[2]	**Gap(%)**	**UV**[1]	**ST**[3]
S5_V52_T7_M1	0.01	0	1.6	0.01	0	1.7	0.00	0	6.8
S5_V77_T8_M1	0.10	3	5	0.03	3	6.5	0.00	3	15
S16_V109_T9_M1	1.2	4	41	0.41	4	235	0.07	4	66
S24_V222_T9_M1	31	9	562	8.6	7	4204	101	8	1828

[1] UV = Unserviced voyages

[2] ST$_{RASO}$ = Total time for routing and speed optimizing in seconds

[3] ST = Solution time in seconds

The results in Table 7.10 show that solving the routing problem with only the highest speed profile available by using the RHH is significant faster than by using the MIP. However, the quality of the solutions are not the same. Here, the sequential routing and speed optimization using MIP performs better. If we compare with the original RHH results, we see that it is only for the largest problem instance that solving the routing with highest speed profile and speed optimize afterwards offer a better solution, however, at the cost of a much longer solution time. Again, we use the modification with the smallest time windows when testing the model. Because of the simplicity of these problem instances, the RHH is not utilized to its full potential, and solving the routing part of the problem with the MIP results in a lower amount of unserviced voyages for the largest problem instance. With increased complexity with modification 2 and 3, the MIP would assumable not be able to solve the routing part of the problem within the time limit. Since the results from using modification 1 illustrates how speed optimization adds complexity to the problem, we have not elaborated further on modification 2 and 3 in this thesis.

The effect of having separate speed variables

When reviewing previous work on similar types of problems, formulations with a separate speed variable for the ballast sailing were not found. A joint speed variable for the voyage and the following ballast sailing was presented in Andersson et al. (2014) and inspired us to extend our formulation to include a separate variable for the ballast sailing. To see the effect of having separate speed variables, the model was modified to have a joint speed variable in the comparison. The MIP model for all problem instances for the first modification was used and the comparison is presented in Table 7.11.

Table 7.11 – Comparison to a modified model with joint speed variables. The gaps reported for the modified model with joint speed variables are calculated using the best bound from the MIP

	Joint speed variables				**Original MIP formulation**			
Instance	**OC**[1]	**Gap(%)**	**UV**[2]	**ST**[3]	**OC**[1]	**Gap(%)**	**UV**[2]	**ST**[3]
S2_V5_T3_M1	1.05500E+5	0.00	0	0.14	1.05500E+5	0.00	0	0.14
S5_V52_T7_M1	3.15082E+8	0.00	0	80	3.15082E+8	0.00	0	17
S5_V77_T8_M1	1.52109E+8	0.01	3	445	1.52057E+8	0.00	3	29
S16_V109_T9_M1	2.58164E+8	0.14	4	10,000	2.57669E+8	0.07	4	10,000
S24_V222_T9_M1	1.52184E+8	673	64	10,000	2.07788E+8	101	15	10,000

[1] OC = Operating costs
[2] UV = Unserviced voyages
[3] ST = Solution time

The quality of the solutions seem very similar, with only a minor difference in the gaps. Here, the gaps are based on the objective value and the best bound of the MIP. However, when focusing on the operating cost, we see that the differences for the problem instances with the same amount of unserviced voyages are significant. For S5_V77_T8_M1, which is solved to optimality for both formulations, the difference is

$$1.52109 \cdot 10^8 - 1.52057 \cdot 10^8 = \$51,954$$

which equals to a difference in percentage of:

$$\frac{1.52109 \cdot 10^8 - 1.52057 \cdot 10^8}{1.52057 \cdot 10^8} = 0.0003 = 0.03\%$$

This means that the formulation with separate speed formulations slightly improves the solution. We also observe a large difference in operating costs for the problem instance S16_V109_T9_M1, but because the model is still running after 10,000 seconds, the solution could still be improved and the results are thus not further elaborated.

For the largest problem instance the operating cost decreases significantly because of the high number of unserviced voyages in the formulation with joint speed profiles, and is therefore not appropriate to compare with the original MIP formulation.

The extension of having separate speed variables gave equal or better solution quality within a shorter amount of time compared to having a joint speed variable. We also believe that if the model was extended to include completion deadlines for each voyage, the use of separate speed variables could be utilized to a greater extent. I.e., a vessel could sail with a high speed profile on the voyage in order to complete the voyage sailing within the time limit, and then choose a lower speed profile to save costs when sailing ballast to the next voyage.

The effect of wider time windows

In reality, shipping companies may operate with wide time windows for the contracted service of a voyage. This section examines the effect of time window widths. For this purpose, the results from the RHH are revisited and sorted according to the problem instances and the modifications in Table 7.12.

Table 7.12 illustrates that wider time windows lowers the total cost, although it increases the solution time significantly. The quality of the solutions are observed to be decreasing when focusing on the gap for the solutions. This is explained by the increased solution space when the time windows are expanded, and the RHH is not able to prove optimal operating costs within the time limit. The cost reduction occurs because voyages may

Table 7.12 – Results from RHH revisited and sorted by each problem instance in order to show the effect of time windows. The smallest problem instance is due to its simplicity omitted from this study.

Instance	**OC**[1]	**Gap (%)**	**UV**[2]	**ST**[3]
S5_V52_T7_M1	3.15149E+8	0.02	0	6.8
S5_V52_T7_M2	3.18215E+8	0.09	0	36
S5_V52_T7_M3	3.17889E+8	0.06	0	40
S5_V77_T8_M1	4.52502E+8	0.10	3	15
S5_V77_T8_M2	6.51637E+8	45	5	75
S5_V77_T8_M3	4.51695E+8	0.72	3	86
S16_V109_T9_M1	6.62067E+8	0.74	4	66
S16_V109_T9_M2	5.55004E+8	1.6	3	121
S16_V109_T9_M3	5.53730E+8	2.3	3	210
S25_V222_T9_M1	1.01610E+9	20	8	1,828
S25_V222_T9_M2	6.28334E+8	4.3	4	3,420
S25_V222_T9_M3	6.26286E+8	4.8	4	5,520

[1] OC = Operating cost
[2] UV = Unserviced voyages
[3] ST = Solution time (s)

be started earlier and can be sailed with a lower speed profile. The servicing vessel does not need to wait for the target date to start sailing the voyage, and utilizes the previous waiting by sailing with a lower speed profile.

For problem instance S16_V109_T9, it can be seen that the RHH has found good quality solutions for all modifications and that the first modification contains one more unserviced voyage in the best bound. This shows how wider time windows could make it possible to service additional voyages. This analysis shows that shipping companies may reduce costs by negotiating wider time windows for its contracted voyages. For example, for the largest problem instance that nearly was solved to optimality for all modifications, S5_V77_T8 the cost reduction is

$$\frac{1.52057 \cdot 10^8 - 1.51464 \cdot 10^8}{1.51464 \cdot 10^8} = 0.0039 = 0.39\%$$

and does not impact the solution much in terms of absolute savings. However, noticeable cost reductions are achieved if this reduction may be applied permanently to the company's total operating costs. Also, by examining Tables 7.6 and 7.5 more closely, we observe that wider time windows lead to fewer unserviced voyages. More flexibility results in a better utilization of the fleet and lower operating costs, and improves the company's accommodation of obligations, i.e. its contracted voyages.

Chapter 8

Concluding Remarks I

In Part I of this thesis we have formulated a mixed-integer program (MIP), an adaptive large neighborhood search (ALNS) and a rolling horizon heuristic (RHH) to solve the fleet deployment problem (FDP) for a Roll-on Roll-off (RoRo) liner shipping company. The FDP consists of assigning vessels to voyages with given time windows and to cover monthly demands. Similarly as in Andersson et al. (2014), speed optimization is integrated in the fleet deployment. Unlike container shipping, cargo segments and deck types are segmented and specifically formulated. The objective when solving the FDP is to minimize costs.

Reviewing previous literature led to the hypothesis that the MIP alone is not capable of solving larger, realistic instances of the FDP within practical time frames due to its complexity. Therefore, two heuristic solving methods were developed. The ALNS uses destroy and repair operators to explore the solution space in order to find good solutions to the problem. The RHH iterates through the planning horizon and solves the problem by dividing it into sub-horizons and sub-problems. A forecasting period is used to take future decisions into account.

A computational study showed that the MIP was able to solve smaller problem instances to optimum within reasonable computing times, but it was not able to find good solutions within 10,000 seconds to larger instances. The ALNS found better solutions than the MIP for the larger instances, but these were not as good or found as fast as for the RHH. The RHH proved superior, both in terms of quality and solution time, in finding solutions to these instances. Wider time windows and multiple speed profile options increase problem complexity, but facilitates more cost-efficient solutions. Our proposed formulation with two separate speed variables for sailing voyages and sailing ballast provided solutions of equal or better quality than having a joint speed variable as used in previous literature. The presented models may be used as a decision support tool for RoRo shipping companies when planning their fleet deployment, and can also be adapted to other liner shipping segments.

Future work

During the work on this thesis we have identified several possible improvements to the models. In this section we address potential areas of future work.

The mathematical model can be extended to include more realistic features. Similar to how time windows define when vessels can start a voyage, a deadline can also be set for the completion of each voyage. This extension may increase the complexity of the problem, but also offer more options and information that could be utilized by planners. Another interesting extension is to include variables that describe when vessels are idle. This could help incorporate more realistic costs in the model, as well as the possibility of reducing operating costs and chartering out vessels for longer idle periods.

Several possible improvements were identified while designing and implementing the presented ALNS. First, the simulated annealing framework should be further analyzed. Reheating could be introduced to diverse the search at larger iteration levels. Second, loads and demands may be given further attention and used to affect the assignment of voyages. Similar to how contracts and remaining monthly demands are ranked in Stålhane et al. (2012), it could be possible to rank voyages according to their balance categories and corresponding accommodated demands. This may help assign voyages more efficiently and reduce the need for space chartering.

As described in Section 7.4, the RHH could in some cases provide undesirable results when a voyage has its upper time window limit in one of the first days of the month. A simple suggestion to prevent this from happening would be to modify the instances and move the latest start date of a voyage away from the first few days of a month. This could be done by either increase or reduce the width of the time windows. Another suggested improvement to the RHH to prevent this issue from occurring is to solve the model several times with different days that define the length of the primary and forecasting periods. As the different modifications of problem instance S5_V77_T8 illustrated, moving the time windows by only one day could have an impact on the solution. Several RHH models with different primary and forecasting period configurations should be run simultaneously to ensure that a potentially higher solution quality is not achieved at the expense of longer solution time.

The presented FDP is solved as a deterministic model with no uncertainty associated with the parameters. Maritime transportation operates in an unstable environment, where planners face uncertainty on a daily basis. Extending the model to include a form of disruption management would provide a more comprehensive decision support. Part II of this thesis is dedicated to this important area.

Part II

Disruption Management in Roll-on Roll-off Liner Shipping

Chapter 9

Problem Description II

This chapter gives an introduction to the challenges that arises when considering fleet deployment in an uncertain environment. First, we discuss how events and uncertainties affect operations in the Roll-on Roll-off (RoRo) segment of liner shipping. Then we discuss how robust planning may better prepare the fleet deployment for disruptive events during the planning horizon. Finally, the possibility of including real-time re-planning for recovering from disruptions is considered. By identifying events, introduce robust planning and incorporate real-time re-planning we aim to present a disruption management decision support model for the case company.

Events and uncertainties in RoRo liner shipping

In Part I, the fleet deployment problem (FDP) for a RoRo liner shipping company is presented and solved by using different solving methods. At this point, all parameters are assumed to have a known value, i.e. the problem can be solved deterministically. However, maritime transportation is constantly affected by events which could affect the underlying parameter values and any ongoing operations. Identifying these types of events and their likeliness of occurring is the first step in building a more realistic and applicable model.

A deterministic approach to solve the FDP would base its parameters on one specific scenario, a set of expected values made from historical data, or another parameter value generation technique used by the company. However, in order to make the model as realistic as possible, the underlying uncertainty in the parameter values must be captured. We must determine how to test the performance of the models with randomly distributed disruptive events occurring during the planning horizon. In this evaluation framework, events will change the values of the input parameters, and it must be determined how the changes will impact the ongoing operation of the fleet.

Adding robustness

Making the model more robust will allow events to occur during the planning period and change any parameter values without having a severe impact on the original plan. There are several ways of making models more robust, but the common consequence is increased cost of the original plan. By for example adding buffer time between two subsequent voyages, the vessel will have to sail at a higher speed, resulting in higher planned costs. This is the cost of making the model more robust, and may pay off if an event occurs and a longer sailing duration of a voyage is experienced. Ideally, a robust model could prevent higher costs when disruptive events occur, but it is necessary to determine the wanted degree of robustness. If the in-advance planning accounts for a large number of events and overestimates their impacts, the plan will most likely be very costly. Hence, it is important to find the optimal degree of robustness in the fleet deployment.

To make a fleet deployment plan more robust, we need to develop a set of robustness measures and include them when solving the models presented in Part I. A literature review may reveal several ways of adding robustness when planning, which may serve as an inspiration when considering robustness strategies for the FDP. It must also be determined how to weight the different strategies if several methods are chosen simultaneously.

Real time re-planning

A different approach of handling disruptions is to alter the original plan in cases when disruptions occur. A new plan that aims to recover the operations and minimize the impact of disruptions will have to be developed. Finding this so-called *recovery plan* is a similar problem to the original FDP, where the objective is to minimize costs. However, the new parameter values caused by disruptive events must be considered. Also, the recovery plan must find a new solution within a reasonable amount of time to be useful for the case company. The solution found needs to be applicable for the case company in terms of implementation. Making many changes to the plan may be expensive or not desirable for other reasons. Therefore, a trade-off between the number of changes resulting from the recovery plan and minimizing costs must be considered.

Summary

In Part II we aim to integrate disruption management in the fleet deployment decision support model presented in Part I for a RoRo liner shipping case company. The FDP consists of assigning vessels to voyages with given time windows and to cover monthly demands. In reality, we must also consider the possibility of disruptive events that could affect the parameter values during the execution of the plan. To include disruption management in the decision support model, we must (1) identify a set of key events and their associated impacts, (2) determine ways of adding robustness when planning, and (3) facilitate the use of recovery strategies when re-planning is found necessary during the execution of plans.

Chapter 10

Literature Review II

In this chapter we review relevant literature in the field of disruption management. First, in Section 10.1 a set of fundamental publications and studies on disruption management is presented. This may help us understand the need for disruption management, its traditional framework, and the previous areas where it has been applied. The presented literature is both shipping-specific and from other relevant fields of research. Then, in Sections 10.2 and 10.3 we review two different approaches of dealing with uncertainties, *robust planning* and *recovery strategies*. Relevant publications on disruption management in the airline industry is also presented, due to similarities and considerably larger attention given in the literature. Finally, a summary stating the lack of research on robust fleet deployment and recovery strategies in the RoRo shipping segment concludes the chapter and provide motivation for the problem studied in part two of this thesis.

10.1 Introduction to Disruption Management

The term *disruption* is used when unforeseen events cause a deviation from an original plan. The performance of the plan may severely be affected by the changes made to the system by the occurred event. Events and their subsequent disruptions are caused by internal as well as external factors. Yu and Qi (2004) present a general framework of disruption management, and structure various sources of disruptions into categories. Amongst these are changes in system environment, unpredictable events, changes in system parameters, changes in availability of resources, new restrictions, uncertainties in system performance and new considerations. Rough weather conditions, union strikes, delivery times, new industry regulations, uncertain time of completion for an important project and new customer priorities are examples from these categories, respectively. In addition, some events which have the potential of causing major disruptions are very difficult or even impossible to anticipate, such as piracy or terrorist attacks.

In the context of liner shipping, Christiansen et al. (2004) state that maritime operations have large operational uncertainty that could cause unforeseen events. Further on, some common events causing disruptions are mentioned, such as port congestion, bad weather at sea, labor strikes, mechanical failures and tidal windows. Notteboom (2006) explores the relative importance of sources of schedule reliability in the liner service schedules, and reveals that port congestion and lower-than-expected port productivity account for a majority of the schedule unreliability on the East Asia-Europe route. See Figure 10.1 below for a more detailed breakdown from the survey conducted by the author. Due to the nature of liner shipping networks, a disruption will cascade through the network and may have consequences for other vessels and future port calls (Notteboom and Rodrigue, 2008). For instance, Notteboom (2006) points out that it may take days or even weeks for terminals in Europe and on the US east coast to recover from major disruptions due to bad weather conditions in the Atlantic Ocean. A disturbance of port operations may have consequences for vessels with upcoming calls at the affected ports. Paul and Maloni (2010) provide a decision support system to help port networks analyze disruption scenarios, where an algorithm routes arriving vessels to ports in order to optimize the use of network capacity.

There are significant economic impacts associated with disruptions in liner shipping, and Kjeldsen et al. (2011) highlight that this has an effect on two fronts. First, there are costs to the shipping company such as increased bunker cost, increased port costs, charter costs of extra ships and cargo space, and intangible costs (e.g., goodwill, loss of customers). As illustrated in Figure 4.1, an increase in service speed results in a substantial increase of fuel consumption. Kjeldsen et al. (2011) exemplify this by showing that increasing the service speed of an 6,600 TEU container vessel from 18 to 24 knots may increase fuel consumption by up to 130 tons per day. With the fuel prices of USD 650 per ton at the time the research was conducted, the higher speed costs USD 84,500 extra for each day it has to be maintained. Secondly, in addition to the cost to the liner shipping company, there are significant costs associated for the customers whose cargo are onboard the vessels subject to disruption. Estimations done by Notteboom (2006) show that a vessel carrying 4,000 TEUs travelling from the Far East to Belgium may lead to extra costs for its customers of at least EUR 57,000 per day it is delayed.

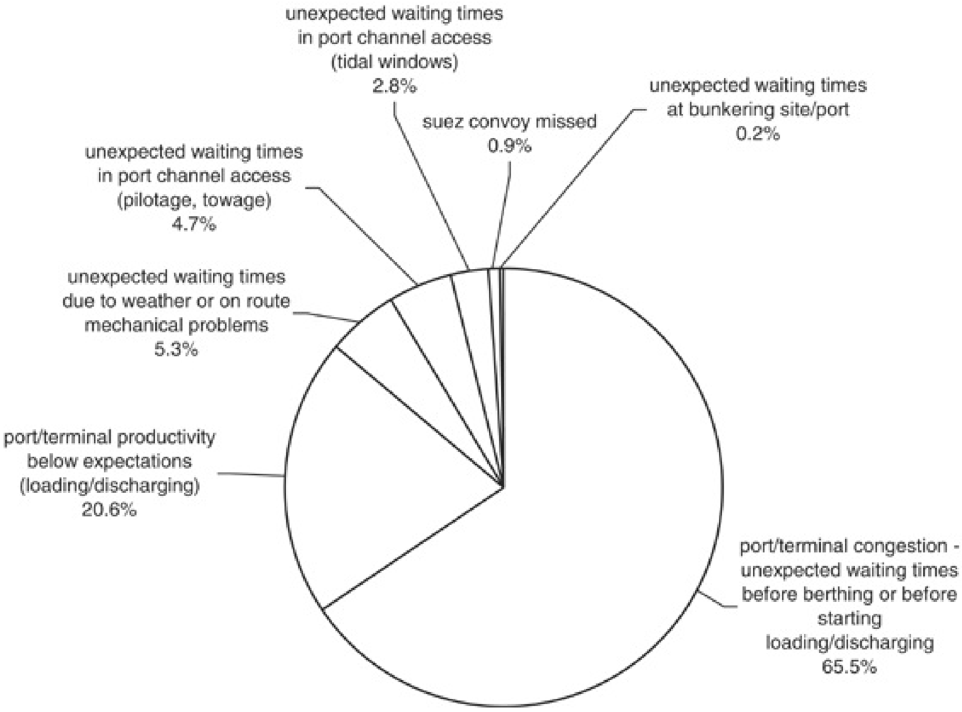

Figure 10.1 – Sources of schedule unreliability on the East Asia–Europe liner shipping route for the fourth quarter of 2004. Source: Notteboom (2006)

Moreover, Yu and Qi (2004) highlight that great efforts have been made in the past several decades to deal with uncertainties through different approaches. The authors classify these approaches into two stages: in-advance planning and real-time re-planning. The purpose of in-advance planning is to find an optimal plan while taking future uncertainties into account. This may be referred to as robust planning or robust optimization. Here, future uncertainties can be modeled by a set of scenarios, and the goal of robust optimization is to generate a plan that is "good" for many of the possible future outcomes (Yu and Qi, 2004). Real-time re-planning aims to revise the original plan previously found whenever needed during the period of execution. This may be when an event occurs and leads to a disruption which causes the original plan to be undesirable.

Uncertainties may arise from various sources, and can be characterized by having different forms, frequencies and degrees of significance. Qi (2015) considers two categories of uncertainties, which are based on the factors of (1) frequency of occurrence and (2) familiarity, and name them *recurring and regular* uncertainties, and *rare and irregular uncertainties*. For the former category, historical data can be utilized to establish models to predict uncertainties. This type of uncertainties rarely causes major implications for an on going plan, because, due to their regularity, their impacts can be proactively incorporated. For the latter category of rare and irregular uncertainties, it is likely that they will occur from time to time. However, they are impossible to quantify from historical data or predict in any way.

To illustrate this, Qi (2015) uses a simple example from reality. If you drive to work in the morning by car, it is natural to plan some extra time in case of traffic jam. However, it is hard (and to a certain point superfluous) to account for the possibility that one day the car may break down. Hence, we must from a certain point rely on real-time decisions, *recovery strategies*, when a disruption occurs. In this example, we may rather take a taxi or the bus, depending on the importance of being on time that morning.

From this, we conclude that a disruption management system for liner shipping should include robustness in the in-advance planning process to account for recurring and regular uncertainties, and a set of recovery strategies for dealing with rare and irregular uncertainties during the execution of the plan. The two following sections will explore relevant literature published regarding robust planning and recovery strategies, respectively.

10.2 Robust Planning

According to Yu and Qi (2004), a robust planning process consists of the following steps:

1. Identifying a set of potential disruptive scenarios
2. Choosing a robustness criterion appropriate for the decision maker
3. Incorporate the above information and measure in planning to generate a robust plan
4. Carry out the plan without change no matter what may happen in the future

In practice, these steps have various degrees of difficulty. For step 1, shipping companies may use historical data to identify relevant disruptive scenarios. However, defining a robustness criterion as stated by step 2 might not be straightforward. For example, Kouvelis and Yu (1997) define one robustness criterion as minimizing the maximum deviation from optimality. Many studies, where only a few concern maritime transportation, have tested a set of different robustness criteria and will be reviewed below.

Further on, Yu and Qi (2004) highlight that if a "good" robust solution is chosen, this plan will not cause any extreme inferior results when executed. However, the other way around is not necessarily true. If no changes are needed during execution, the plan may have been too conservative and likely very costly for the company. Due to this, it is not always wise to follow step 4, and a robust plan may in practice indeed be subject to changes throughout its horizon.

When exploring ways to incorporate uncertainty in the in-advance planning process, it is natural to come across stochastic optimization. However, as pointed out by Bertsimas and Sim (2003), there are two main difficulties associated with this approach: (1) Creating scenarios that capture the uncertainty distribution, and (2) the size of the resulting stochastic optimization model and the associated computational challenges. As we have observed in Part I of this paper, the fleet deployment problem already is computationally challenging as a deterministic problem. Therefore, robust planning appears as an excellent alternative to stochastic optimization in creating robust fleet deployment plans.

As mentioned above, a set of publications have used various criteria in the making of robust in-advance plans. Christiansen and Fagerholt (2002) present a ship scheduling problem concerned with the pickup and delivery of bulk cargoes within given time windows. Here, ports are closed at night and during weekends, and loading/discharging may take several days. In other words, ships will spend a lot of time idle in port, and the total time depends on the ships' arrival times. The paper's objective is to make robust schedules that are less likely to result in ships staying in ports over the weekend, with weather and port service times treated as uncertain. The problem is solved by imposing penalty costs for arrivals at "risky" times, i.e. close to weekends. Robust schedules are achieved at the sacrifice of an increase in planned transportation costs, though hopefully decreasing the expected cost of performing the plan.

In Halvorsen-Weare and Fagerholt (2011), robust solutions to a supply vessel planning problem are created. The model is based on a previous study, Halvorsen-Weare et al. (2012), which resulted in substantial savings for the case company. However, due to disruptions, the planners experienced that the deterministic plans presented often were difficult to execute in real life. Robust solutions to the problem may help avoid unplanned and costly means of satisfying critical demand. In order to make schedules robust, Halvorsen-Weare and Fagerholt (2011) suggest three different ways of adding slack to the voyages and vessel schedules: (1) Reward each day a vessel is idle, (2) reward each vessel that has at least an idle day each week, and (3) reward vessels that sail no more than two voyages each week. A computational study is conducted to evaluate which criteria that are creating the best robust solutions in a simulation framework. A strategy combining various robustness measures is found most effective.

The same authors also study vessel routing and scheduling under uncertainty in the LNG business in Halvorsen-Weare et al. (2013). In the considered real-life problem, an LNG producer is responsible for transportation from production sites to customers all over the world. The aim is to create vessel schedules that are more robust against uncertainty in sailing times due to changing weather. Four robustness strategies are proposed: (1) Adding extra sailing time, i.e. for a trip that usually takes 30 days, it can be planned to take 32 days, (2) target inventory level in storage tanks at ports, (3) target accumulated berth use and (4) combined strategies of these. The robustness strategies are evaluated in a simulation-optimization framework. Here, a re-route optimization procedure are called whenever plans are needed to be adjusted (i.e., whenever a customer cargo cannot be picked up on the planned day). The re-optimized plan is used in the remaining schedule, and does not include any robustness strategies. The findings were that the combined robustness strategy gave the best results, and that there is a significant improvement potential by adding the proposed robustness approaches.

In Sokol et al. (2015), a robust maritime inventory routing problem with time windows and stochastic travel times are studied. The authors propose a two-phase solution approach that considers a sample set of major disruptions and associated recovery solutions. In the first phase, two planning strategies to generate robust routes are proposed: (1) Evenly allocate idle time, and (2) separate deliveries with a minimum time requirement. A multi-scenario construction heuristic is used in the second phase to obtain good feasible solutions. The computational results presented reveal that the solution procedure leads to robust solutions with lower expected costs.

There are a few additional articles that considers robustness in maritime planning problems, such as List et al. (2003), Zeng et al. (2010), Kjeldsen et al. (2011) and Alvarez et al. (2011). It is worth noticing that Kjeldsen et al. (2011), which mainly consider rescheduling, mention that a buffer in order to secure schedule reliability in liner shipping can be included in two different ways. First, it can be included as an extension of the port call, i.e. a scheduled port call is longer than what would normally be required. This is mostly relevant for unreliable ports, in terms of variability and congestion. Second, a buffer can also be included by increasing sailing times.

Also, several studies of similar robustness problems are relevant, such as from the fields of production scheduling and the airline industry. Sitompul and Aghezzaf (2008) focuse on safety stock in a production scheduling problem, and propose and discuss an alternative model for aggregate production planning when periodic demands are uncertain. Clausen et al. (2010), Ehrgott and Ryan (2002), Barnhart (2009) and Smith and Johnson (2006) are some examples from research done in the airline industry which focuses on robustness in crew and aircraft schedules. Buffer time and station purity are two common robustness strategies within this field. For further in-depth studies, the reader is referred to these presented articles.

10.3 Recovery Strategies

Due to the degree of uncertainty in maritime operations, shipping companies will sooner or later experience the occurrence of disruptive events. Despite having a robust initial plan, the updated system parameters may cause it to be infeasible or very undesirable from an economic perspective. In these situations, it is necessary to have a set of recovery strategies in order to minimize the impact of the disruption on the remaining horizon of the plan.

In addition, revising a plan is associated with some deviation costs (Yu and Qi, 2004). These costs could be sunk fees, waste of raw material, hiring or using expensive personnel or, perhaps more importantly in certain regions, the loss of customers' goodwill for waiting and delay. It is important to take these deviation costs into account when generating the new plan. Without considering these, Yu and Qi (2004) emphasize that the recovery solution may include too many unwanted or unrealistic changes to the plan, making it difficult to implement.

In Andersen (2010), the network transition problem in container liner shipping is studied, i.e. the process of moving assets operating in an existing network to a new one. An adaptive large neighborhood search is developed and shown to produce competitive solutions. Andersen (2010) points out that network transition and disruption recovery share a common structure, and that the concept of network transition faces the same challenges as a recovery scenario. The only difference is the time horizon and the number of assets involved. Therefore, the solution approach is proposed to be equally capable of addressing disruption recovery problems. However, Andersen (2010) highlights that the recovery problem will typically be smaller and have more restrictive requirements to computation times. A similar problem, the liner shipping fleet repositioning problem (LS-FRP), is studied in Tierney and Jensen (2012) and extended with cargo flows in Tierney et al. (2014). Tierney and Jensen (2012) point out that the network transition problem studied by Andersen (2010) lacks cost saving activities like Sail-On-Service opportunities, empty equipment flows and slow steaming. A mixed-integer program including the two first activities is presented and found effective on a set of smaller real world problem instances.

Kjeldsen et al. (2011) highlight that the developed solution method in Andersen (2010) is only capable of addressing the rescheduling problem if the disruption has already happened. Further, a mathematical model for simultaneous rescheduling of vessels and cargo in container liner shipping in the event of past, present, and future disruptions is presented. Given the list of disruptions, the planning period and the involved vessels and ports, the objective of the model is to construct a set of vessel schedules and cargo routings that allow resumption of the scheduled service at the end of the planning period while minimizing costs (Kjeldsen et al., 2011). In order to solve the problem, a large neighborhood search is developed and found computationally effective. A set of recovery strategies are also rendered from Notteboom (2006), namely reshuffling port calls, cancel port calls, 'cut and run', deploy other vessels to take over, and increase speed to catch up. A cut-and-run strategy means to depart before all (un-)loading moves have been performed. Deploying other vessels can be illustrated through chartering a ship merely for the purpose of recovering from a major disruption (Notteboom, 2006).

In Brouer et al. (2013), the vessel schedule recovery problem (VSRP) for container liner shipping is studied. The publication, which is based on work done in the master thesis of Dirksen (2011), presents a MIP model for handling disruptions in liner shipping. The VSRP evaluates a current disruption scenario and selects the action that balances increased bunker consumption, impact on cargo and customer service level by using multi-criteria optimization in the objective function as presented in Ehrgott (2006). A computational study conducted on four real-life cases reveal potential cost savings of up to 58 % compared to real-time solution chosen by the case company.

Further, Qi (2015) takes the ideas of Brouer et al. (2013) and develop a model that handles recovery of multiple vessels in a network after a major disruption. Two different decisions are handled by the model: vessel routing and speeding decisions, and the container flow. The model extends the work done by Brouer et al. (2013), which assume that containers will be transported by their original vessels in the event of a disruption, by including the container rerouting. The presented formulation in Qi (2015) includes two inter-correlated multi-commodity network flow problems, and the author points out that this may be hard to solve for large-scale problems. No application or computational study of the mathematical model is presented.

The literature presented above represent the few research papers concerning disruption recovery in maritime transportation and liner shipping we were able to find. In addition to these, several articles dealing with similar problems in the airline industry were found relevant. Set against the liner shipping industry, a significant deal of research have been done on disruption management within the airline industry. As early as in 1984, Teodorović and Guberinić (1984) presented a problem where the objective is to minimize the total passenger delay if one or more aircraft become unavailable. Since then, numerous publications have been done. Clausen et al. (2010) perform a general survey of the field, and divides the conducted research into three main areas: Aircraft recovery, crew recovery and integrated flight and passenger recovery. Kjeldsen et al. (2011) point out that integrated recovery of passengers and flights are similar to the problem of rescheduling ships and cargo in liner shipping. In the following paragraph, a few articles from the airline industry concerning this problem are reviewed.

In Barnhart (2009), a set of regular actions performed by airline controllers when a disruption occurs are listed. Amongst these are delaying flight departures, cancelling flight legs, rerouting or swapping aircraft, and re-accommodating disrupted passengers. Dienst et al. (2012) present and compare two models for the aircraft recovery problem. When solving the models for real life problems, it is natural to believe that an unlimited number of changes to the schedule cannot be implemented last minute. Therefore, the paper uses two different mechanisms for reducing the number of changes done when recovering. These are (1) a unit penalty cost for each change, and (2) introducing protection arcs in the model, i.e. arcs associated with a certain "bonus". Both were found effective, but (2) was found to be computationally faster. In Sinclair et al. (2014), a large neighborhood search heuristic to solve the integrated aircraft and passenger recovery problem is presented.

10.4 Summary

The increasing amount of work done within the field of disruption management illustrates the importance of dealing with uncertainties. Initially lagging behind the airline industry, it is clear that the liner shipping industry has been given more attention in the later years, e.g. Andersen (2010), Kjeldsen et al. (2011) and Brouer et al. (2013). However, what has been done in the airline industry and partly the containerized network liner shipping business cannot be directly applied to the RoRo segment in liner shipping.

The reviewed literature shows that there is a lack of research within this field. With a tough competitive environment, substantial customer expectations and a high degree of uncertainty in operations, a need for disruption management system, both in-advance planning and recovery planning, for RoRo shipping is apparent.

Chapter 11

Events and Disruptions in Liner Shipping

As presented in Section 1.3 and suggested in Chapter 9, identifying possible events and disruptions is the first of three important aspects when handling uncertainties. In this chapter we elaborate on the possible events and consecutive disruptions that may occur during the operation of a RoRo liner shipping fleet. Informal talks with representatives from the case company enlightened the daily operations and where disruptions usually occur. We have also studied relevant literature to get a comprehensive overview of disruptive events in the RoRo liner shipping segment. In this chapter we first present common events and their potential consequences for the case company, before describing how these events are simplified and included in our models.

In Chapters 1 and 10, some events are mentioned as possible reasons for disruptions. Here we present a set of events that could occur for the case company in different stages of the operations. For the purpose of this thesis, these events are simplified and divided into happening (1) in ports, and (2) when sailing voyages or sailing ballast. This segmentation complies with the findings of Notteboom (2006), where schedule unreliability in liner shipping is studied (see Figure 10.1 and Section 10.1). Port and sailing events are presented in Sections 11.1 and 11.2, respectively. Finally, in Section 11.3 we describe how these events are considered in our model.

11.1 Ports

As presented in Notteboom (2006), port and terminal congestion alone stand for over 65 % of schedule unreliability on the studied shipping route in the fourth quarter of 2004. The case company's vessels visit several ports during the planning period when servicing voyages. A voyage consists of two or more port calls, where the vessels refuel, load and discharge their cargo, and may undergo basic maintenance. During these activities, it is according to the case company not unusual that disruptive events occur. Some of these identified events are presented below.

Bunkering

Usually, it is the local port workers that conduct the work of rebunkering the vessels during port visits. The case company has experienced several incidents where port workers claim they have bunkered the vessel with more fuel than what it is indicated on the vessel's instruments. Due to the large volumes of fuel being bunkered to these types of vessels, even small deviations will have severe associated costs. Disagreements related to the amount of bunkered fuel have previously lead to tiresome negotiations and caused vessels to wait until a resolution is found between the parties. This often requires a third party, called a surveyor, to enter the negotiations and find a reasonable agreement. Waiting for the parties to find a resolution may cause delays from a couple of hours and up to as much as several days.

Congestion

As presented in Chapter 1, the volume transported by the maritime sector has significantly increased in the later years. This requires a higher cargo handling capacity in ports. According to Vernimmen et al. (2007), the increase in port capacity has not been sufficient to match this cargo volume development. Hua (2013) discusses how the fast growth in vehicle flows has imposed China to license more ports for import and how port space issues cause problems for shipping companies. Due to RoRo cargo being space-intensive, it requires a lot of time and effort to relocate the cargo if needed. Figure 11.1 shows the RoRo terminal facilities in Le Havre, France, and illustrates the substantial space-intensity of RoRo cargo. If the port has a tight schedule and a vessel is delayed, it could affect other port calls and disrupt port operations. With an increasing transportation demand and insufficient expansion of port capacity, these situations are more likely to occur. Notteboom (2006) states that the rising port volumes and capacity constraints in many ports around the world mean that berth availability on arrival in a port is not always guaranteed when an allocated time slot has been missed. If a vessel misses its slot, it will have to wait until a slot is open, and this could take up to weeks in the busiest ports.

Figure 11.1 – RoRo terminal facilities in Le Havre, France. Source: Grand Port Maritime du Havre

Strike

The maritime industry is, as every other industry, at risk of having disagreements with its employers. In major cases this could lead to work stoppage in the form of labor strikes in ports. A potential consequence of this is limited operation or even a shutdown of the port, which may cause delays for shipping companies' operations. According to representatives from the case company, the impact will typically vary between two to four days if the port is open with limited activity, and could be significantly longer if the port is shut down completely. For instance, in February 2015, the ports of Los Angeles and Long Beach faced a huge backlog after a long dispute between port workers and their employers. The leader of the Port of Los Angeles suggested it would take as much as three months "to get back a sense of normalcy" (NBC News, 2015). Figure 11.2 shows how vessels were piling up as the strike haltered the US West Coast port operations in the winter of 2015.

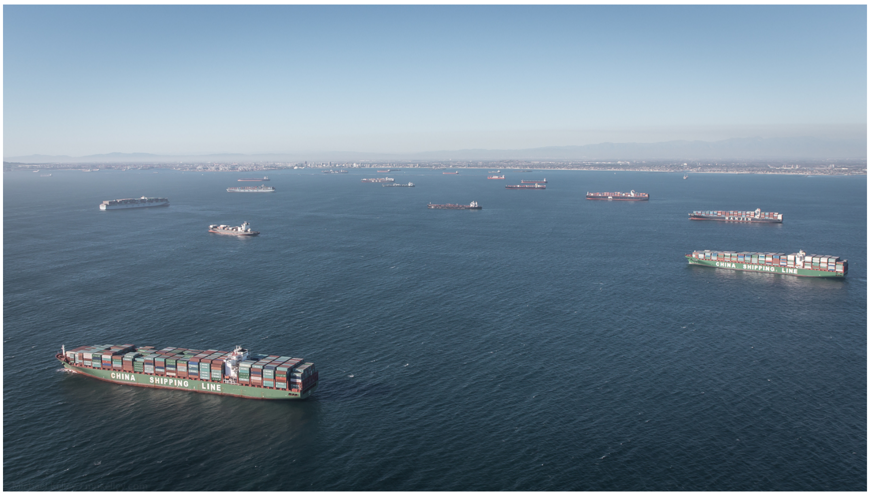

Figure 11.2 – Vessels waiting outside US West Coast ports during the labor strike in February, 2015. Source: VR WORLD

11.2 Sailing

Sailing in open water includes dealing with the elemental forces and their associated uncertainties. In addition, some routes involve passageways, rivers and tidal windows the vessels have to use in order to reach certain ports. Figure 11.3 shows how sailing times between Rome, Italy, and Bergen, Norway, for different types of vessels are distributed. It is clear that, even for this relatively short route, sailing times are widely spread out. In this section we present some disruptive events the case company's vessels may face when sailing voyages or sailing ballast.

Rough weather conditions

Sailing voyages in open water and on transatlantic routes includes the risk of being hit by bad weather. Certain areas have seasons with rough conditions, for instance the typhoon season in Asia and the hurricane season in North America. Bad weather may result in vessels moving slower than planned, which may cause delays. In Halvorsen-Weare and Fagerholt (2011), the weather impact for supply vessels is considered. The significant wave height in rough weather is highlighted in the study as the critical factor that limits the operation of the supply vessels. Although the RoRo vessels studied in this thesis are larger than the supply vessels in the study, the impact of the wave height in rough weather is also considered as applicable to the RoRo vessels. Table 11.1 shows how different wave heights impact the speed for the supply vessels in Halvorsen-Weare and Fagerholt (2011).

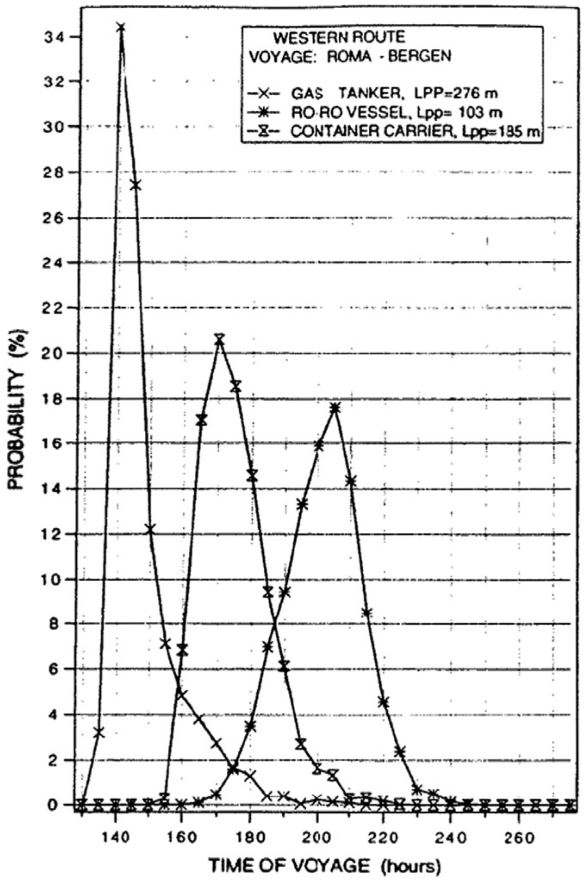

Figure 11.3 – Probability distributions for sailing times between Rome (Italy) and Bergen (Norway) for gas tankers, RoRo vessels and container ships. Source: Kauczynski (1994).

Rough weather may also cause vessels to wait near a port before starting a voyage. At some ports, e.g. the port of Cape Town, it is according to Vernimmen et al. (2007) not unusual that vessels become delayed for several days while waiting for the weather to improve. Vessels waiting near ports are not considered in Table 11.1, but the impact of increased voyage duration for shipping companies is assumed to be similar.

Strong current in rivers and tidal windows

Some of the ports operated by the case company are located in river deltas, where local tides cause the water current to vary. If the vessels are to sail counter-current, the sailing duration may increase and possibly cause delays. Also, Notteboom (2006) mentions that the largest vessels need to take into account tidal window restrictions in the access channels of some ports. Although a series of deepening programs have substantially widened these windows, ports like Antwerp, Hamburg and Bremerhaven have areas with

Table 11.1 – The impact of rough weather on sailing speed of supply vessels. Source: Halvorsen-Weare and Fagerholt (2011).

Weather state	Wave height [m]	Reduced sailing speed [kn]
1	< 2.5	0
2	$2.5 < 3.5$	0
3	$3.5 < 4.5$	- 2
4	≥ 4.5	- 3

shallow waters that may complicate the design of a liner service schedule and could lead to changes in the order of port calls.

Breakdowns

Vessel breakdowns may happen due to many different reasons, for example mechanical failure or bad fuel quality. According to representatives from the case company, breakdowns are regular events, especially for older vessels, and these events could result in reduced speeds and longer sailing times. Depending on the extent of a breakdown, consequential delays could be between a couple of hours to several weeks.

Congestion in passageways

Sailing maritime passages like the Suez Canal and the Panama Canal are for some of the voyages vital in order to be able to sail the voyage within the planned duration. Some passageways and canals are very busy and could impose the vessels to wait in turn before sailing through. When passing through the canals, vessels sail in convoys and the vessels need to book their place in one of these convoys in advance. If a vessel arrives late at the entrance of the canal and misses its convoy, an additional waiting time of up to 12 hours may occur (Notteboom, 2006).

11.3 Integrating Events into our Model

The disruptive events presented above are all regular threats to the original plan for the case company. When integrating disruption management into our model, we base the possible disruptive events on the ones that are described in the previous sections.

Differences in probabilities and impacts make it appropriate to divide the disruptive events into the two categories suggested: Port events and sailing events. In each of these categories, all the respective events presented above are merged. However, each of the categories have different degrees of disruption and will therefore have several possible impacts. As a simplification we have only considered disruptions of voyage sailings and not the ballast sailings between the voyages.

Port disruption

The probability for a port disruption to happen for each port call is based on estimates made by representatives from the case company. As presented in the previous section, a port disruption could cause a delay between a couple of hours and up to several weeks. In our model we have decided that each port disruption will lead to Υ^P days of waiting time before the next voyage can start. A vessel is affected by the port disruption if it is currently sailing a voyage with the disrupted port as the end port.

Constraints (4.12) are the only constraints involving the relationship between the end time of a voyage and the start time of the next voyage. To incorporate the effects of port disruptions into the mathematical model, the parameter Υ^P is added to these constraints. The resulting constraints are shown below in 11.1.

$$y_{vriqj}(t_{ri} - t_{qj} + \Upsilon^p + \sum_{s \in \mathcal{S}_v} (T^B_{vriqjs} x^B_{vriqjs} + T_{vris} x_{vris})) \leq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}^P_v. \tag{11.1}$$

The port disruption parameter Υ^P represents the extra waiting time caused by a port disruption. The set of arcs that are affected by port disruptions for a vessel v is represented by $\mathcal{A}^P_v$. The set of disrupted arcs is defined by connecting the nodes in the set of port disrupted voyages $(r,i) \in \mathcal{N}^P_v$ and the set of all voyages $(q,j) \in \mathcal{N}_v$. The constraints (11.1) are linearized as before as shown in Appendix A.2, and are used in the same way as originally in the mathematical formulation given by (4.1)-(4.30).

Sailing disruption

The sailing disruptions are included in a similar way as the port disruptions. The probability for a disruptive event happening during sailing is also based on considerations made by representatives for the case company. Because the duration of the voyages will vary, the degree of impact of the disruptions are modeled by multiplying the sailing duration with the parameter Υ^S_{ri}. We consider the remaining sailing time of the voyage when adding Υ^S_{ri}. This calculation is shown in Algorithm 11.1.

Constraints (4.12) are the only constraints involving voyage sailing times. These constraints are modified for the voyages affected by sailing disruption as shown below.

$$y_{vriqj}(t_{ri} - t_{qj} + \sum_{s \in \mathcal{S}_v} (T^B_{vriqjs} x^B_{vriqjs} + \Upsilon^S_{ri} T_{vris} x_{vris})) \leq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}^S_v. \tag{11.2}$$

The sailing disruption parameter Υ^S_{ri} represents the additional sailing time, in percentage, of the affected arcs $\mathcal{A}^S_v$. As for port disruptions, the set of arcs disrupted by sailing disruption consists of the arcs connecting the nodes in the set of sailing disrupted voyages $(r,i) \in \mathcal{N}^S_v$ and the set of all voyages $(q,j) \in \mathcal{N}_v$. The parameter Υ^S_{ri} is determined by the remaining length of the voyage and a parameter for the maximum impact of a sailing disruption $\Upsilon^{S,MAX}$. Since the time of sailing the disrupted voyages increase, the costs of sailing these voyages need to be increased equivalently. This is incorporated by multiplying C_{vris} in the objective function with Υ^S_{ri}. The constraints are linearized as before as shown in Appendix A.2 and used in the same way as originally in the mathematical formulation given by (4.1)-(4.30).

Algorithm 11.1. Pseudo Code for Calculating the Sailing Disruption Parameter Υ^S_{ri}

1: Input: disrupted voyage (r,i); vessel v sailing voyage (r,i); maximum impact of sailing disruption $\Upsilon^{S,MAX}$; current day n in the planning period; start time of the voyage t_{ri}; sailing speed profile values of the voyage x_{vris};
2: Set duration of voyage $\tau_{ri} = 0$;
3: Set remaining voyage duration ratio $\phi_{ri} = 0$;
4: **for all** $(s \in \mathcal{S}_v)$ **do**
5: $\quad \tau_{ri} = \tau_{ri} + T_{vris} x_{vris}$
6: **end for**
7: Calculate $\phi_{ri} = ((t_{ri} - n) + \tau_{ri})/\tau_{ri})$;
8: Set $\Upsilon^S_{ri} = 1 + \Upsilon^{S,MAX} \phi_{ri}$;
9: **return** Υ^S_{ri}

Figure 11.4 shows the effect of a sailing disruption on voyage (r,i), here illustrated by the event of bad weather. The sailing time of this voyage is extended in accordance with the resulting sailing disruption parameter Υ^S_{ri}. This results in a delay before the next voyage (q, j) can be started, showed by the shift in the start time of voyage (q, j).

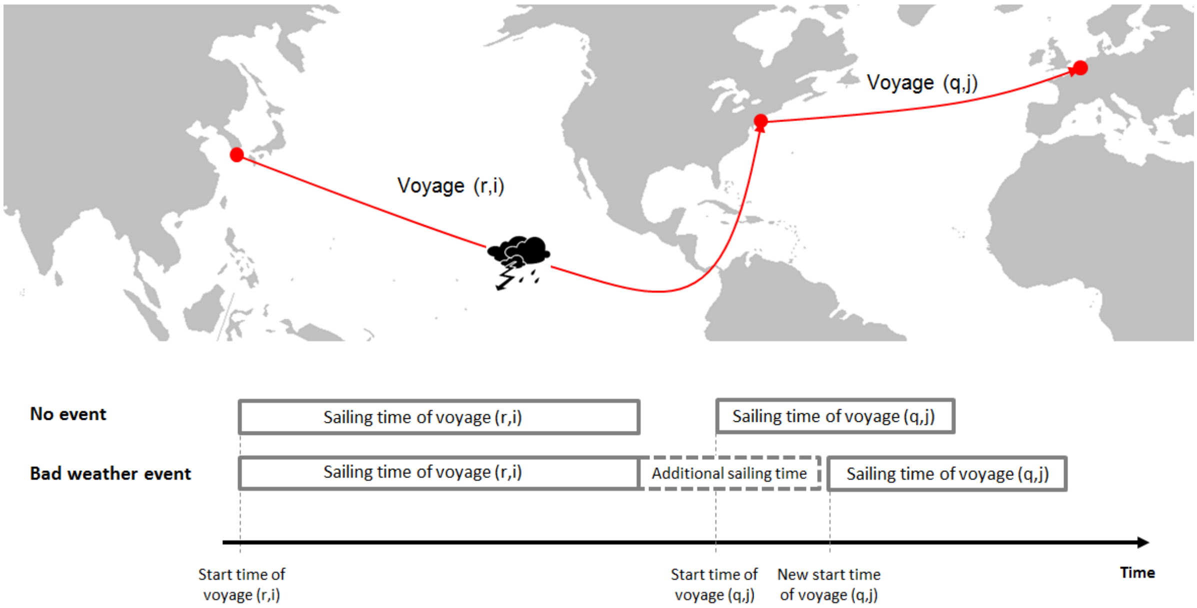

Figure 11.4 – The consequences for a vessel's schedule in the event of a sailing disruption. The resulting extra sailing time of voyage (r, i) causes a shift in the start time of the subsequent voyage (q, j).

Simultaneous sailing and port disruptions

It is possible that a voyage may be simultaneously affected by a sailing disruption and a port disruption. The way of incorporating sailing and port disruptions into the program are then combined. As a simplification, a voyage may only be affected by at most one sailing disruption and one port disruption.

Chapter 12

Robust Fleet Deployment

In this chapter we propose measures of incorporating robustness, *robustness strategies*, when generating fleet deployment plans for the RoRo liner shipping case company. We first state and illustrate the motivation behind the suggested measures in Section 12.1. Then, the additional mathematical implications of the measures for the model are presented in Sections 12.2 - 12.5. Finally, in Section 12.6 we suggest and draw an outline of a simulation framework for evaluating the proposed measures of adding robustness.

12.1 Adding Robustness

The MIP model defined by the mathematical formulation (4.1)-(4.30) in Section 4.2 is capable of solving smaller real-life instances of the FDP for the RoRo liner shipping case company, while heuristics have been shown effective for solving larger problem instances in Chapter 7. However, the solutions obtained so far by using these solving methods may become difficult to execute in practice, as they do not take uncertainties in the parameter values into consideration. In this chapter we propose a set of robustness strategies to make deployment plans better prepared for disruptions. The suggested robustness strategies include the following measures, that work as extensions to the mathematical model presented in Section 4.2:

1. Adding extra sailing time
2. Reward early arrivals
3. Penalizing risky start times

The idea behind these extensions is to create more robust solutions when regarding the possible events described in Chapter 11. The presented measures are motivated by similar ways of adding robustness suggested and shown successful in the reviewed relevant literature in Section 10.2.

It is important to remember that despite any changes made to the mathematical formulation by adding these measures, they are merely artificial and may help a solution method to consider more robust solutions. The extra long sailing times introduced in Section 12.2 prepare the vessel schedules for potentially longer sailing durations in practice. The rewards and penalties included in Sections 12.3 and 12.4 give incentives to make vessels arrive before the start of voyage time windows and avoid late starts, respectively. We must remember that in reality the shipping company is never paid or charged for any revenues or costs associated with these measures. Therefore, the objective values obtained when solving these models are mutually non-comparable. To evaluate the performance of a robust model, other comparison methods must be applied. A simulation framework for evaluating the quality and robustness of solutions obtained with the different robustness measures is suggested in Section 12.6 and thoroughly described in Chapter 13.

12.2 Adding Extra Sailing Time

As presented in Halvorsen-Weare et al. (2013), planning some extra sailing time for each voyage is a straight-forward way of introducing slack in the vessel schedules. This can, for example, imply that a voyage that usually takes 20 days at cruising speed is planned to take 22 days. This means that tight vessel schedules will be found infeasible if they do not tolerate this extra sailing time, and another deployment setup will be chosen. When considering the planned extra sailing times against the actual sailing times of the voyages, the result is that we have introduced slack in the vessel schedules. This is illustrated in Figure 12.1. For the solutions with added sailing time (Figure 12.1b), the difference between planned and actual sailing times ensures that there will always be added slack to the solutions. However, this can result in a vessel not being able to service the same sequence of voyages as in the original solution. This can be observed from Figure 12.1b, where the extra sailing time of voyage 2 limits the vessel from taking voyage 3 as in Figure 12.1a.

As pointed out by Halvorsen-Weare et al. (2013), it is important to notice that this robustness strategy may cause problems when the fleet of vessels is close to or is being

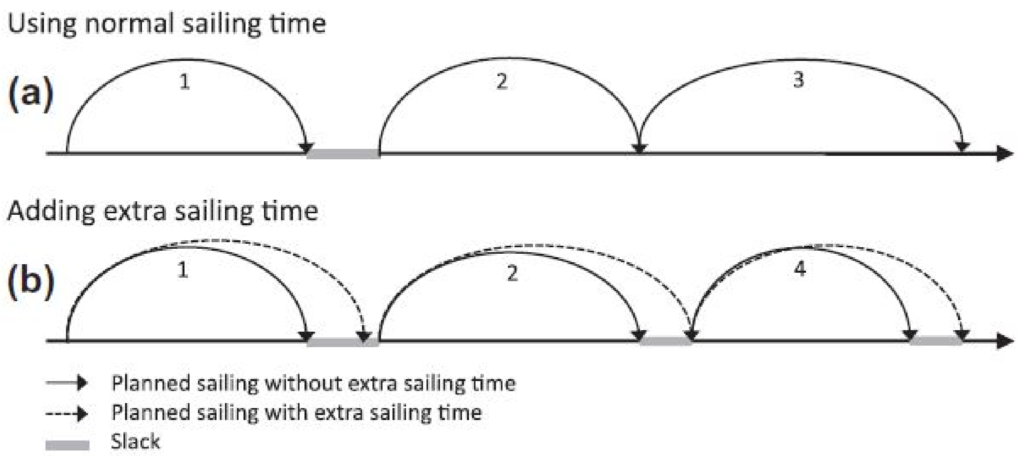

Figure 12.1 – Illustration of a vessel schedule without (a) and with (b) added extra sailing time. In solution (a), the vessel sails voyages 1-2-3, while in (b) the vessel sails voyages 1-2-4. The extra sailing time in (b) ensures slack in the solution, but renders the original schedule infeasible. Source: Halvorsen-Weare et al. (2013).

fully utilized, i.e. the schedules obtained by using original sailing times have little or no slack. By adding extra sailing times to these instances, some voyages will be set as unserviced despite originally being able to cover all voyages with vessels from the company's own fleet. This implies that the robustness strategy of adjusting sailing times must be used with caution, and that the obtained solutions should be compared with the ones obtained by using original sailing times. In cases where additional unserviced voyages are found, the adjusted sailing times should be reduced until these unserviced voyages are eliminated.

Mathematical implications

Constraints (4.11) and (4.12) are the only constraints involving sailing times. These are revisited below.

$$y_{o(v)ri}(t_{o(v)} - t_{ri} + \sum_{s \in \mathcal{S}_v} T^B_{o(v)ris} x^B_{o(v)ris}) \leq 0, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v, \tag{4.11 revisited}$$

$$y_{vriqj}(t_{ri} - t_{qj} + \sum_{s \in \mathcal{S}_v} (T^B_{vriqjs} x^B_{vriqjs} + T_{vris} x_{vris})) \leq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v. \tag{4.12 revisited}$$

In order to add robustness, we adjust the sailing times T^B_{vriqjs}, $T^B_{o(v)ris}$ and T_{vris} by multiplying them by a robustness parameter Ω. An Ω greater than 1 will increase the sailing times. The resulting constraints are linearized as before as shown in Appendix A.2 and used in the same way as originally in the mathematical model given by (4.1)-(4.30).

12.3 Rewarding Early Arrivals

Rewarding properties that introduce slack in solutions have commonly been used in the reviewed literature. For example, Halvorsen-Weare and Fagerholt (2011) present several possible ways of favoring robustness in a solution by adding a robustness profit in the objective function. Here, the proposed properties subject to rewards are (1) each day a vessel is idle, (2) each vessel that has one idle day during the week, or (3) each vessel that sails no more than two voyages during the week. In this section we suggest a similar approach: Rewarding when a vessel arrives before the start of the time window, E_{ri}, of the next assigned voyage.

Mathematical implications

To extend the mathematical formulation (4.1)-(4.30) to reward early arrivals, we need to introduce a few additional variables and parameters. Now, let t^R_{ri} be the number of days rewarded for early arrival before starting voyage (r, i). It is likely that a few days of slack spread out are significantly more valuable than a large period before a single voyage. To avoid this, we propose an upper limit of rewarded days, R^{MAX}, for each early arrival. Further on, let δ^R_{ri} be 1 if an arrival at voyage (r, i) qualifies for a reward, and 0 otherwise. The reward given per time unit of early arrival is denoted as Λ^R. Because we are cost-minimizing, this parameter takes on negative values.

Figure 12.2 illustrates how rewarding early arrivals work in practice. If a vessel arrives at the start of voyage (r, i) before the start of the time window E_{ri}, any slack up to R^{MAX} days will be awarded with a factor of Λ^R (see scenario (a)). Any arrivals after E_{ri}, shown by scenario (b), do not qualify for any rewards.

Now, let us formulate the additional constraints we need to model the rewards of early arrivals:

$$y_{vriqj}(E_{qj}\delta^R_{qj} - t^R_{qj} - t_{ri} - \sum_{s\in\mathcal{S}_v}(T^B_{vriqjs}x^B_{vriqjs} + T_{vris}x_{vris}) + M^R_1(1-\delta^R_{qj})) \geq 0,$$
$$v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v. \quad (12.1)$$

$$y_{o(v)ri}(E_{ri}\delta^R_{ri} - t^R_{ri} - t_{o(v)} - \sum_{s\in\mathcal{S}_v}(T^B_{o(v)ris}x^B_{o(v)ris}) + M^R_2(1-\delta^R_{ri})) \geq 0, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v \quad (12.2)$$

Constraints (12.1) and (12.2) trigger the amount of time rewarded for each voyage. These expressions are non-linear and are linearized in (12.6) and (12.7) .

$$t^R_{ri} \leq R^{MAX}\delta^R_{ri}, \qquad (r,i) \in \mathcal{N}. \quad (12.3)$$

Constraints (12.3) limit t^R_{ri} from exceeding the maximum number of days that are rewarded. The number of rewarded days is set to zero if δ_{ri} is set to zero by (12.1) or (12.2).

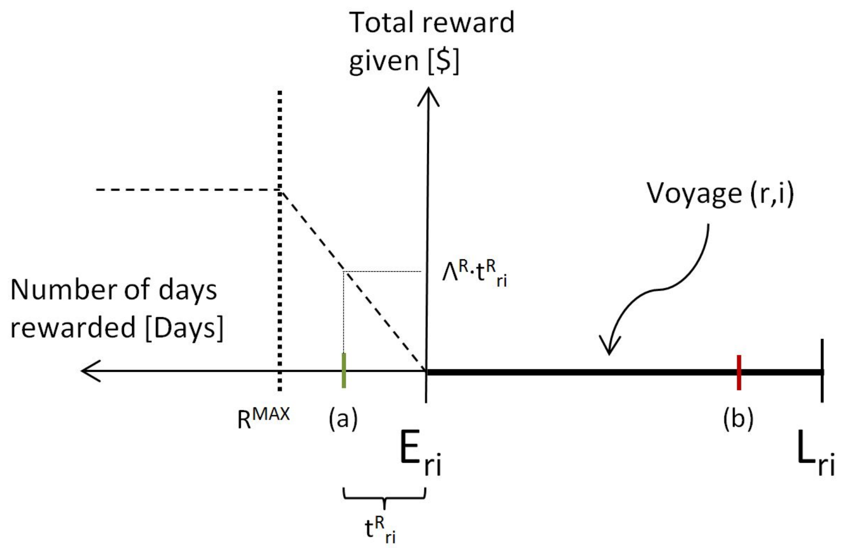

Figure 12.2 – Illustration of (a) rewarding and (b) not rewarding an early arrival. In scenario (a), the vessel arrives at the start of voyage (r,i) t^R_{ri} days before the time window starts, which results in a reward of $\Lambda^R \cdot t^R_{ri}$. In scenario (b), the vessel arrives late in the time window and no reward is given.

$$t^R_{ri} \geq 0, \qquad (r,i) \in \mathcal{N}. \tag{12.4}$$

Constraints (12.4) ensure t^R_{ri} is non-negative. However, in practice this will never happen as it will result in a penalty rather than a reward.

$$\delta^R_{ri} + y^S_{ri} \leq 1, \qquad (r,i) \in \mathcal{N}. \tag{12.5}$$

With constraints (12.5) we avoid that rewards for early arrival at voyage (r,i) are given if the voyage is unserviced.

Linearizing constraints (12.1) and (12.2) respectively, we get:

$$\begin{aligned} M^R_3(1-\delta^R_{qj}) \geq t^R_{qj} + t_{ri} + \sum_{s\in\mathcal{S}_v}(T^B_{vriqjs}x^B_{vriqjs} + T_{vris}x_{vris}) \\ - E_{qj}\delta^R_{qj} - M^R_1(1-y_{vriqj}), \qquad v\in\mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v. \end{aligned} \tag{12.6}$$

$$\begin{aligned} M^R_4(1-\delta^R_{ri}) \geq t^R_{ri} + t_{o(v)} + \sum_{s\in\mathcal{S}_v}(T^B_{o(v)ris}x^B_{o(v)ris}) \\ - E_{ri}\delta^R_{ri} - M^R_2(1-y_{o(v)ri}), \qquad v\in\mathcal{V}, (r,i) \in \mathcal{N}_v. \end{aligned} \tag{12.7}$$

Sufficiently large values of M^R_1 to M^R_4 are calculated in Appendix A.3.

To allocate the appropriate reward, the objective function (4.1) is extended with the additional term (12.8):

$$\sum_{(r,i)\in\mathcal{N}} \Lambda^R t^R_{ri} \tag{12.8}$$

Because Λ^R is negative, any positive values of t^R_{ri} will reduce overall costs.

12.4 Penalizing Risky Voyage Start Times

A similar approach to rewarding particular system properties that provides robustness as in Section 12.3 is, on the other hand, to penalize characteristics of a solution that reduce robustness. This method can be effective if certain solutions are significantly more exposed to changes if an event occurs than others. For example, in Christiansen and Fagerholt (2002), where a pickup and delivery ship scheduling problem with multiple time windows is studied, robust solutions are created by penalizing risky arrivals. Here, risky arrivals are considered as arrivals that are close to the end of harbor opening hours, i.e. close to nights and weekends, because it may result in a ship staying idle much longer than necessary. In the studied problem, Christiansen and Fagerholt (2002) consider weather and service times in ports as uncertain. In this section we suggest a similar approach for creating robust fleet deployment for the RoRo liner case company.

Due to possible events in ports and at sea, sailing times are uncertain. This means that it is not always possible to comply with planned start times. If an event causes a significant delay for a vessel v at a voyage (r, i), the vessel's schedule is already tight, and the next voyage (q, j) has a planned start time t_{qj} close to the end of its time window L_{qj}, a delay d_{qj} is likely to occur. To avoid these situations we can penalize start times close to the end of time windows. This will alter vessel schedules or increase planned sailing speeds, which both are measures to increase slack in the solution and thereby its robustness.

Mathematical implications

To extend the mathematical formulation (4.1)-(4.30) to penalize risky start times, we need to introduce additional variables and parameters. Now, let t^P_{ri} be the number of days penalized for risky start time of voyage (r, i). For each voyage we apply an upper limit P^{MAX}_{ri} to the number of days penalized, which is the smaller of a global upper limit P^{MAX} and the width of the voyage's time window $|L_{ri} - E_{ri}|$. The penalty given per time unit of risky start time is denoted as Λ^P.

Figure 12.3 illustrates how risky start times are penalized. If a vessel plans to start a voyage at the time given by (a), no penalty is given. However, if (b) is the planned start time, this is within the later part of the time window. We classify this as a risky start time, and a penalty with a factor of Λ^P is suffered.

When calculating P^{MAX}_{ri}, we use the following expression:

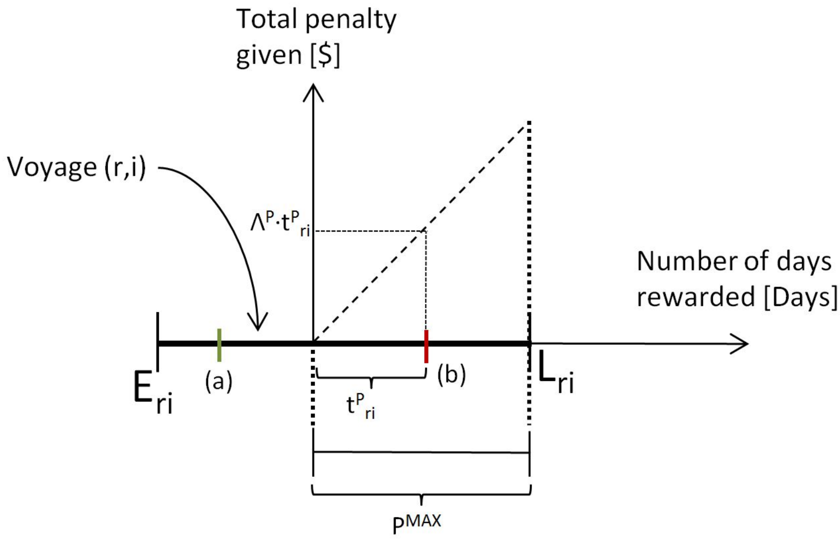

Figure 12.3 – Illustration of not (a) penalizing and (b) penalizing the start time of a voyage. In scenario (a), the vessel is planned to start voyage (r, i) more than P^{MAX} days before the end of the time window. This start time is not considered risky and is therefore not penalized. In scenario (b), the start time of voyage (r, i) is t^P_{ri} days into the risky area of start times for the voyage, which results in a penalty of $\Lambda^P \cdot t^P_{ri}$.

$$P^{MAX}_{ri} = min \left\{ L_{ri} - E_{ri}, P^{MAX} \right\}, \qquad (r, i) \in \mathcal{N}. \tag{12.9}$$

To model the penalizing of risky start times, the original model defined by (4.1)-(4.30) is extended by adding the following constraints.

$$t_{ri} - d_{ri} - (L_{ri} - P^{MAX}_{ri}) - t^P_{ri} \leq 0, \qquad (r, i) \in \mathcal{N}. \tag{12.10}$$

Constraints (12.10) set the number of days penalized for risky start time of voyage (r, i) to be greater than or equal to the start time of the voyage minus the time when penalties start occurring. We also subtract any delay associated with voyage (r, i).

$$0 \leq t^P_{ri} \leq P^{MAX}_{ri}, \qquad (r, i) \in \mathcal{N}. \tag{12.11}$$

Constraints (12.11) set the appropriate range of t^P_{ri}.

Also, to allocate penalties, the objective function (4.1) is extended with an additional term (12.12):

$$\sum_{(r,i)\in\mathcal{N}} \Lambda^P t^P_{ri} \tag{12.12}$$

12.5 Combining Robustness Measures

In Halvorsen-Weare et al. (2013), a combined strategy of robustness measures were found the most effective in producing quality solutions for an LNG ship routing and scheduling problem. It is possible that a combination of the measures presented in Sections 12.2, 12.3 and 12.4 may produce solutions of better quality than the individual measures on their own for the FDP as well.

The proposed combined strategy is a combination of all the three robustness measures presented. This means that the constraints (12.3)-(12.7) and (12.9)-(12.11) are all added to the model formulation from Section 4.2, together with adjusting the sailing times by Ω. The objective function (4.1) is extended with the additional terms:

$$\sum_{(r,i)\in\mathcal{N}} \Lambda^R t^R_{ri} + \sum_{(r,i)\in\mathcal{N}} \Lambda^P t^P_{ri} \tag{12.13}$$

12.6 Evaluation of Robustness in Deployment Plans

To evaluate the presented robustness strategies, a simulation framework is proposed. This framework must be able to simulate the planning horizon of fleet deployment instances, and randomly apply realistic events causing disruptions in the plan. The robustness measures may help the initial fleet deployment to be better prepared for these types of unforeseen events. A set of possible events is presented in Chapter 11, and the simulation framework used in evaluating the quality of solutions is described in depth in Chapter 13.

As presented in Clausen et al. (2010), robustness can be divided into (1) absorption robustness and (2) recovery robustness. We evaluate absorption robustness of solutions by testing how slack in the schedules absorb the effect of disruptive events. This means that no recovery actions should be enabled. However, we consider this as an unrealistic scenario, for two particular reasons. First, it is unlikely that a shipping company simply will stand and watch its vessels become majorly delayed. Second, *slow-steaming*, i.e. sailing at low speeds, can be considered as a form of slack, because vessels may speed up to recover from disruptions. By not allowing recovery actions, including speed adjustments, this slack is not appreciated. Therefore, we also test a combination of absorption and recovery robustness of the various strategies by allowing speed adjustments as recovery actions when simulating.

A plan designed with an emphasis on recovery robustness will more likely be able to make use of recovery actions to mitigate the effects of disruptive events. We evaluate recovery robustness of solutions by calling a re-optimization recovery procedure described in Section 13.2 every time a certain triggering condition is met. This procedure may utilize speed adjustments, start time adjustments, cargo reallocation and even voyage swaps to recover from a disruption.

In order to evaluate solutions, we must determine a set of evaluation criteria to consider. Conservativeness, risk profile, flexibility and different customer bases and their associated preferences may affect what companies consider as a quality solution. Some companies may accept less robust solutions with a lower expected cost, despite high probabilities of delays. Other companies may value costlier but more robust solutions to reduce potential delays as much as possible and satisfy challenging customers. In Chapter 14, we compare different types of solutions under various measures.

Chapter 13

A Simulation-Optimization Framework for Evaluating Solutions

In this chapter we propose a simulation-optimization framework to (1) evaluate solutions obtained by applying alternative robustness strategies and (2) evaluate the effectiveness of recovery actions. The framework simulates over the planning horizon of the FDP and introduces uncertainty by randomly adding events. These events may affect the sailing times of voyages and the port times for vessels. The framework combines simulation with optimization by calling a re-planning recovery procedure when certain conditions during the simulation are met. In Section 13.1 we give an overview of the program. Then, a presentation of the re-planning recovery procedure follows in Section 13.2.

13.1 The Simulation Program

The purpose behind the presented simulation-optimization framework is to evaluate a solution in an environment where uncertainties may impact the original plan. The solutions we evaluate can be obtained by solving an FDP instance with the deterministic mathematical model presented in Chapter 4, or with the addition of the robustness strategies presented in Chapter 12. We define a solution as the decisions made by solving the FDP, i.e. (1) which voyages to be sailed by which vessel, (2) at what speed, (3) at what time, (4) in which sequence, (5) the cargo placement at vessel decks, (6) how much cargo to space charter to complement the shipping company fleet, and (7) voyages which cannot be sailed by the company's own fleet, if any.

Simulation programs have to some extent been used in previous relevant literature to evaluate the robustness and quality of solutions. An optimization and simulation framework is used to generate robust schedules in a supply vessel planning problem in Halvorsen-Weare and Fagerholt (2011). The solution method consists of three steps, where (1) all candidate voyages the vessels may sail are generated, (2) candidate voyages are simulated and assigned a robustness measure, and (3) a voyage based model is solved with these robustness measures assigned to voyages. Different probabilities define the weather states, where each state has a different effect on the sailing speed of vessels. Further, in Halvorsen-Weare et al. (2013) LNG vessel routing and scheduling under uncertainty is studied. A simulation model with a recourse action procedure is used to evaluate different robust solutions. The latter study and the simulation framework it presents has been an important inspiration to the program we have developed.

The simulation program we propose introduces uncertainty to the fleet deployment by randomly adding events during the planning horizon. The type of events that can be added and how they affect vessel schedules are described in Chapter 11. For each problem instance we use a set of predefined events, *scenarios*, generated in advance. Together with a solution, these act as an input to the simulation program.

In addition to randomly imposed events, the simulation program features a re-planning procedure. This feature allows for recovery actions, i.e. the possibility of adjusting the plan. These types of actions may help mitigate the effect of events. Whenever triggered, the fleet deployment plan is re-optimized with respect to future schedules, and this new plan is used in the remaining days of the current simulation or until the recovery procedure is triggered again. The trigger for calling the procedure could, for instance, be defined as (1) whenever an event occurs, (2) when delays are anticipated by following the current plan, or (3) when a certain amount of delay is experienced. The selection of trigger for the re-optimization procedure is further studied in Section 14.3. In reality, it is natural to continuously consider alterations to the plan as events appear. The key factor for the case company is to meet its obligations, i.e. service its contracted voyages on time, and not which vessel that sails them.

When considering recovery actions, we must keep in mind that the extent of changes made to the plan may not be desirable from the company's perspective. A slightly more costly plan may involve significantly fewer changes and may hence be more desirable. Therefore, we track the number of changes to the original plan resulting from recovery actions during a simulation. The particular changes we track are voyage swaps between vessels. These types of plan alterations are undesirable because vessels may have different setups and crew requirements. Any voyage swaps imply that new arrangements must be made by the case company. Therefore, we track planned changes and not only incurred changes. Voyage swaps are recorded when a new plan generated by the re-planning recovery procedure includes vessel schedules were voyages have been altered.

A flow chart for the developed simulation program is presented in Figure 13.1. The solution we send as an input to the simulation is obtained by solving the FDP for a given problem instance. We also predefine the event scenario, i.e. the course of events to be added during the simulation. A simulation starts on day $t = 0$, and we iterate through each day t in the planning horizon to see if an event happens at that day. If an event occurs on day t, we update vessel schedules with any impacts of the disruptive event. This means that sailing parameters are adjusted if any voyage or ballast sailings are affected, and extra port time is added in case some vessels are on their way to the relevant port(s). These new schedules are then evaluated, and the re-planning procedure is called if the new schedules satisfy a certain criteria. The re-planning procedure treats all parts of the schedules up to day t as fixed, and tries to mitigate any occurred disruptions as effectively as possible. This is done by performing a set of appropriate recovery actions, such as speeding up vessels, swapping voyages between vessel schedules, or reallocate loads. After re-optimizing, the vessel schedules are updated with the effects of the recovery actions. Then, we check if there are more days left in the planning period. If there are, we update $t = t + 1$ and move on to the next day. If this is the last day, we terminate the simulation. Finally, the resulting solution and a set of simulation statistics are collected as outputs from the simulation.

After a simulation has ended, we use the output to calculate the planned versus incurred costs of the resulting vessel schedules, the total voyage delays and the number of recovery actions performed. These measures are used to assess the input solution of the simulation with respect to robustness and quality. After simulating a set of different solutions, e.g. obtained from using various robustness strategies, through a set of different event scenarios, we can calculate average values and standard deviations, and evaluate the quality of the input solutions. Depending on its preferences, the company can decide which solutions that meet their interests in the best way. For some companies, incurred costs are the obvious most important measure. For others, where client relationships are crucial, the voyage delays must be minimized to keep clients happy. The simulation results may show that one robustness strategy is effective in meeting one company's criteria, while another strategy excels in meeting different criteria.

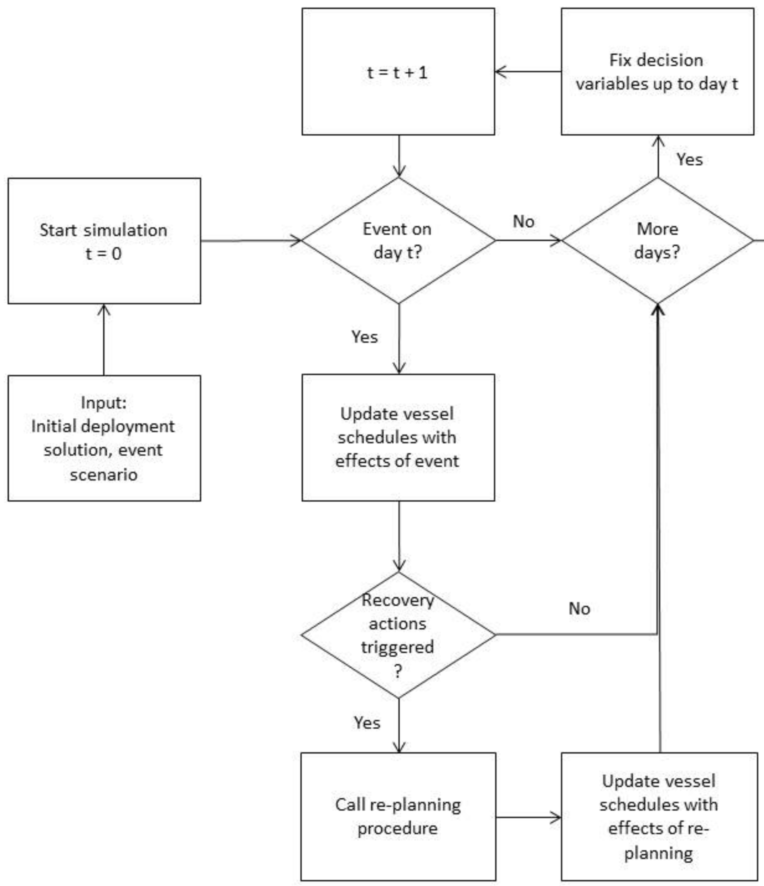

Figure 13.1 – Flow chart for the simulation program.

13.2 The Re-planning Recovery Procedure

The re-planning recovery procedure allows for recovery actions to mitigate the impacts of disruptions as effectively as possible. This is a re-optimization procedure triggered every time the current vessel schedules satisfy a given condition. We include the possibility of recovery actions in the simulation program in order to mimic the real-life planning process as accurately as possible. However, the most effective actions are very hard to identify in a realistic planning environment, especially with many vessels and voyages to consider and limited time available for actions. The presented re-planning procedure may serve as a decision support tool for planners in such situations.

The re-planning recovery procedure consists of solving a constrained version of the mathematical program defined in Section 4.2. The objective is to create a new solution of vessel schedules that minimizes the costs given the updated conditions set by occurred events. This means that vessel speeds can be adjusted, cargo can be reassigned and voyages can be altered between vessels. The need for re-planning appears in real life situations when an event has caused significant disruptions, and immediate actions must be initiated. Therefore, the recovery strategy selected for the procedure must be able to find good solutions quickly. The problem to be solved when the procedure is called is a constrained version of the original program since we only consider the remaining part of the planning horizon. This is given by the day t the re-planning procedure is called.

When solving the recovery re-optimization program, we also include any robustness measures that were used to generate the simulated solution. This is done for two particular reasons. First, it is assumable that additional future contracted voyages have been confirmed and should be added to the problem at the time the recovery procedure is called. We should also include robustness in the schedules when planning for these. Second, if we do not consider the initial robustness measure used to generate the plan, we may loose the benefits it previously provided. Any slack in the vessel schedules may disappear by, e.g., allowing vessels to sail at lower speeds. Without the original slack, upcoming events may cause significant delays.

Algorithm 13.1 shows the pseudo code for the re-planning recovery procedure. Let us assume the re-planning procedure is called on day t. The input to the procedure are the previous solution and all of its contents y^p. Now, all variables associated with commenced voyages up to day t are fixed in the program. This restricts vessels from adjusting their past actions. Next, we solve the remaining partly fixed MIP defined by (4.1)-(4.30), the fixing of variables and any updated parameter values resulting from events. The obtained solution y^t contains the new vessel schedules we use in the continuation of the simulation program.

Algorithm 13.1. Pseudo Code for the Re-planning Recovery Procedure

Input: previous solution y^p, current day in simulation program t;
Set of voyages $\mathcal{N}$;
Voyage start time in previous solution y^p, t_{ri};
for all voyages $(r,i) \in \mathcal{N}$ **do**
    if voyage start time $t_{ri} \leq t$ **then**
        Fix assigned vessel, start time, speed and cargo load of voyage (r,i) given in y^p;
    end if
    for all voyages $(q,j) \in \mathcal{N}$ **do**
        if ballast sailing $((r,i),(q,j))$ has been initiated **then**
            Fix ballast sailing;
        end if
    end for
end for
Solve the optimization program defined by (4.1)-(4.30), the current robustness strategy, the fixed variables and any updated parameters;
Set solution obtained $= y^t$;
return y^t;

Chapter 14

Computational Study II

In this chapter we present a computational study where the robustness strategies given in Chapter 12 are tested in the simulation framework presented in Chapter 13. The strategies are tested both with and without recovery actions. First, in Section 14.1, we provide a schematic overview of the testing procedure and a description of the selected problem instances, test settings and parameter values. Then, in Section 14.2 we compare the results of simulating solutions obtained by applying robustness strategies to the planning procedure. Further, in Section 14.3 we test how different conditions for triggering the recovery re-optimization procedure affect the performance of solutions. Finally, in Section 14.4 we test what combination of robustness and recovery actions that performs best overall, and thoroughly discuss these results.

14.1 Test Settings and Schematic Overview

In this section we describe the selected problem instance and test settings, provide a schematic overview of the testing procedure and state the chosen parameter values.

Problem instance and test settings

We solve problem instance S24_V222_T9_M2 in all testing and simulations performed in this chapter. This problem instance was first introduced in Section 7.1, and is considered as the most realistic, both in terms of size and time window widths, of the problem instances available. S24_V222_T9_M2 consists of 24 vessel, 222 voyages and a time horizon of nine months. A summary of the problem instance characteristics is given in Figure 14.1.

Table 14.1 – Problem instance characteristics

Instance	Vessels	Voyages	Months	Speed profiles	Cargo segments
S24_V222_T9_M2	24	222	9	3	4

The simulation program and the model for creating initial solutions were coded in Java and run in the Eclipse Luna integrated development environment for Java Developers Release 1 (4.4.1) programming interface. All optimization problems, i.e. the generation of initial plans and in the re-planning recovery procedure, were solved using Gurobi Optimizer version 6.0 (Gurobi Optimization, 2014) in the Java interface. All of the models were run on a cluster of computers with HP DL165 G6, 2 x AMD Opteron 2431 2.4 GHz, 24 GB of RAM, and 164 GB SAS 15000rpm.

Schematic overview and parameter values

A schematic overview of the testing procedure is presented in Figure 14.1. In Step 1 we generate a set of scenarios to be used in the simulations. A scenario includes information about when events are happening, the type of events (sailing or port), and where they take place. If we simulate solutions on randomized scenarios, we cannot compare the performance of these solutions directly. The number of events, where they occur and the magnitude of the impact may vary for each simulation and will result in different operating costs and variations in incurred delays. Therefore, we generate a set of scenarios prior to the simulations and use these as an input when simulating the solutions. This means that events are randomly generated beforehand and not during the simulation program. This allows us to compare how different initial plans perform on the same scenario of events.

Table 14.2 shows the probabilities of an event occurring at a given sailing or port on a given day, and the impact of the event. Sailing and port events, their associated implications and the parameters Υ^P, Υ^S_{ri} and $\Upsilon^{S,MAX}$ are further described in Section 11.3.

Table 14.3 shows the average number of sailing and port events that is scheduled to occur during the planning horizon in the scenarios used in the simulations. The corresponding

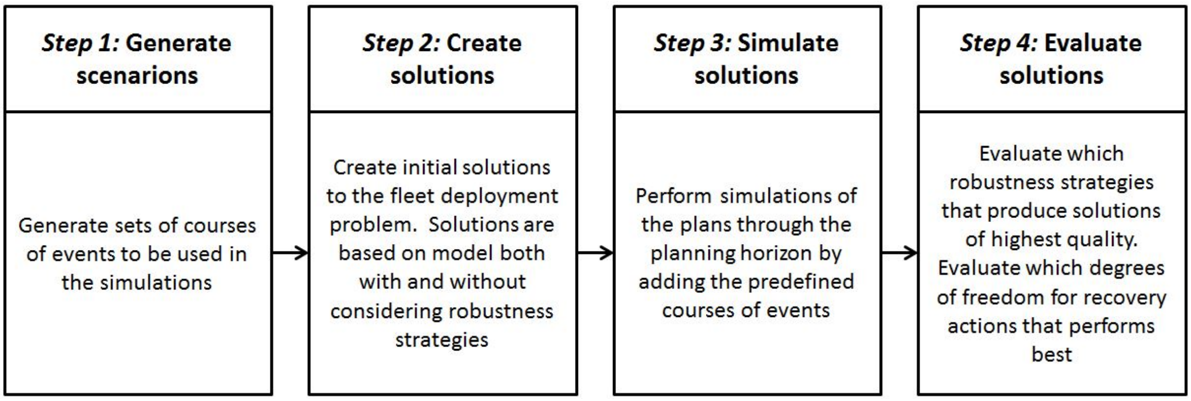

Figure 14.1 – Schematic overview of the evaluation procedure.

Table 14.2 – Events probabilities and impacts

Event	**Probability**	**Parameter**	**Impact**
Port	1.00 %	Υ^{P}	2 [days]
Voyage sailing	2.00 %	$\Upsilon^{S,MAX}$	1.10 [-]

standard deviation is also given. These numbers do not define how many events that affect vessel schedules, as zero, one or several voyages may be traversing the same trade route or towards the same port simultaneously. Consequently, no, one or several disruptions are recorded.

Table 14.3 – Event scenarios characteristics

Event	**Average number of events**	**Standard deviation (%)**
Port	25.3	14.32
Trade route	49.7	12.00

Further, in Step 2 initial fleet deployment solutions to the problem instance are created. Based on the findings in Part I we use the RHH presented in Chapter 6 to create the initial solutions. Different deployment plans for the planning horizon may be obtained by using various robustness strategies.

In Step 3 we simulate the initial solutions created in Step 2 through the planning horizon. For each simulation we add a predefined scenario of events. When simulating solutions, we iterate through each day in the planning horizon and check the current scenario if an event happens at that day. The re-planning recovery procedure may or may not be activated, depending on what we aim to evaluate. Initial testing showed that ten simulations on different event scenarios were sufficient to prove statistical significance with a 95 % confidence level for most combinations of robustness strategies and simulation settings. This is further elaborated in Appendix B.1.

Finally, in Step 4 the performance of solutions are evaluated. In this step we examine how different robustness strategies perform in terms of solution robustness and quality, and how calling the re-planning recovery procedure affects the performance and outcome of initial plans. When evaluating solutions we primarily consider planned costs, simulated costs, total days of delay and voyage swaps.

14.2 Evaluating Robustness Strategies

Four different robustness strategies for creating solutions to the FDP are presented in Chapter 12. Together with applying no robustness measure at all, i.e. solving the model as it is presented in Section 4.2, five different approaches are evaluated in the developed simulation-optimization framework. Table 14.4 gives a description of these approaches and how they are denoted in the remaining of this computational study.

Table 14.4 – List of evaluated robustness strategies

Robustness strategy	Description
BASIC	Mathematical model as presented in Section 4.2
AST	BASIC with adjusted sailing times
REA	BASIC with rewarding early arrivals
PRST	BASIC with penalizing risky start times
COMBINED	A combination of AST, REA and PRST

When generating initial solutions with the given robustness strategies, we must ensure that the solutions obtain the same number of unserviced voyages to be comparable. Due to the added complexity of robustness measures, the RHH may require extra time to find solutions satisfying this premise. We use the following criteria when creating initial solutions with the RHH for the BASIC, AST, REA, PRST and COMBINED strategies:

1. Optimal solution.
2. Best solution found after 10,000 seconds.
3. First solution found with the lower bound of unserviced voyages.

Table 14.5 shows the parameter settings for the robustness strategies which were found effective during initial testing. These settings are used when creating all initial solutions in this computational study. The sailing time adjustment parameter Ω was originally intended to have a higher value, but testing showed that even a slight value increase would increment the number of unserviced voyages and render solutions inferior. Further, similarly to when solving the deterministic model in Part I, we do not allow to plan for any voyage delays, i.e. D^{MAX} remains zero. When simulating, however, delays may occur. As in Section 4.1, the cost of delays are denoted as C^P, and this penalty cost is set according to suggested values from the case company. In addition to presenting the robustness parameter settings, D^{MAX} and C^P are revisited in Table 14.5.

Table 14.5 – Robustness parameter values

Parameter	Description	Value
Ω	Sailing time adjustment	1.02
R^{MAX}	The maximum number of days rewarded for early arrival	2
Λ^R	Reward given per day of early arrival	-$150,000
P^{MAX}	The maximum number of days penalized for risky arrival	2
Λ^P	Penalty suffered per day of risky arrival	$100,000
D^{MAX}	Number of days a voyage can be planned to be delayed	0
C^P	Daily delay penalty cost	$200,000

For the COMBINED robustness strategy we slightly scale down the presented parameter values to incorporate a weighting effect. This was found to be working well during the initial testing. The revised parameter values for the COMBINED strategy is presented in Table 14.6.

Table 14.6 – Revised parameter values for COMBINED robustness strategy

Parameter	Description	Value
Ω	Sailing time adjustment	1.01
Λ^R	Reward given per day of early arrival	-$50,000
Λ^P	Penalty suffered per day of risky arrival	$50,000

Numerical results

Table 14.7 shows the planned operating costs of solutions obtained by the different robustness strategies as a percentage of the BASIC planned operating cost. These costs are calculated as the costs of the solutions given by the objective function (4.1) when the costs of any unserviced voyages have been deducted. Also, any costs or revenues associated with robustness rewards or penalties are excluded in the planned operating costs.

The results in Table 14.7 shows that the solution based on the REA robustness strategy has the largest planned operating costs. This strategy rewards early arrivals, which encourage the solver to speed up the vessels to collect rewards. When the reward itself is deducted, the planned high speeds result in higher operating costs. Similarly, by increasing the sailing time of the voyages, the AST solution makes it necessary to speed up the vessels to complete voyages before the end of the time windows of the subsequent voyage. Further, as the PRST robustness strategy penalizes late starts in the time window, previous voyages are given more time to be completed before they are affected by the robustness measure compared to the REA strategy. Thus, the PRST strategy encourages vessels to operate at lower sailing speeds than the REA strategy. This might explain why the PRST solution has the lowest planned operating costs of the considered robustness strategies. Finally, the COMBINED solution is a combination of the previous methods, and gives, understandably, a planned operating cost in the middle of these. The overall increase in costs for the robustness strategies shows the initial costs of incorporating robustness.

Table 14.7 – Average planned costs for different robustness strategies over ten simulations. The planned costs of solutions obtained with the robustness strategies AST, REA, PRST and COMBINED are expressed as % of the BASIC planned costs

	BASIC	**AST**	**REA**	**PRST**	**COMBINED**
Plan. cost (%)	100.00	102.84	105.00	101.42	103.67

Table 14.8 shows the average simulated costs and average total days of delay for solutions based on the strategies BASIC, AST, REA, PRST and COMBINED. The simulations of initial solutions are conducted with two different settings. For the simulations denoted by NN, we do not allow speed adjustments or any other recovery actions. This setting makes it possible to observe how slack in the solutions can absorb the impact of disruptions directly. For the simulations denoted by ON, we allow speed adjustments to mitigate

the impacts of disruptive events. These speed adjustments may be conducted every day when simulating. The intention behind this setting is to mimic how we assume shipping companies respond to events in reality. Further, this ensures any robustness that comes with sailing voyages at low speeds are appropriately considered. This mix of absorption and recovery robustness is further discussed in Section 12.6. The re-planning recovery procedure is not enabled when simulating with the NN or the ON settings. The results are based on ten simulations with the same event scenarios, and the resulting t-values and corresponding confidence levels are presented in Appendix B.1.

The simulated costs in Table 14.8 are all given as a percentage of the planned cost for the initial BASIC solution. Compared to the planned costs in Table 14.7, the simulated costs capture the consequential costs of uncertainties, i.e. the impacts on the vessel schedules of adding disruptive events. These impacts potentially include longer sailing times, higher speeds, extra space chartering of cargoes and, perhaps most importantly, the costs of voyage delays.

Table 14.8 – Average simulated cost and total days of delay for the different robustness strategies. The simulated costs are expressed as % of the planned costs for the BASIC initial solution.

	BASIC		**AST**		**REA**		**PRST**		**COMBINED**	
	SC[1](%)	**D[2]**	**SC[2](%)**	**D[2]**	**SC[2](%)**	**D[2]**	**SC[2](%)**	**D[2]**	**SC[2](%)**	**D[2]**
NN[3]	155.02	604	133.72	329	117.06	113	143.38	454	116.15	118
ON[4]	121.50	197	118.93	142	114.86	82	115.40	112	113.41	77

[1] Simulated cost in percent of planned cost for the BASIC initial solution.
[2] Total days of delay.
[3] No speed adjustments or any other recovery actions are allowed.
[4] Speeds may be adjusted every day during the simulation. Other types of recovery actions are not allowed.

From the results in Table 14.8 we observe that the simulated costs are higher than the planned costs, meaning that disruptive events have occurred during the planning period and caused extra costs. Also, the results show that the simulated costs when speed adjustments are allowed are substantially lower than when no recovery actions are enabled. By allowing speed adjustments, delays can to some extent be mitigated. This means that there are some slack included in the solutions by having vessels sailing at lower speeds.

By incorporating robustness strategies when planning, we observe a significant improvement in performance in terms of simulated costs and amount of delays. Despite higher planned costs than the BASIC solution, as presented in Table 14.7, all of the robustness solutions have lower average simulated costs. It is also worth noticing that the two solutions with the highest planned costs, COMBINED and REA, are the solutions that provide the lowest simulated costs, both when simulating with the NN and the ON settings.

Further, the simulated costs with the NN setting show a greater variety compared to the ON setting, and could imply that the robustness strategies that performed worse with the NN setting are able to better exploit the opportunity to adjust speeds. This is especially noticeable for the PRST solution, that has the highest simulated costs and amount of delays of the robustness strategies with the NN setting, but it is able to perform nearly equally as well as the solutions based on the REA and COMBINED strategies with the ON setting. A possible explanation is that the PRST strategy only encourages small slacks in the vessel schedules. When allowing speed adjustments the PRST strategy is able to utilize the option of increasing the speed of the vessels sailing the disrupted voyages during the simulation to avoid delays. The small differences in simulated costs and delays for the REA solution can be explained similarly. Here, the solver is motivated to increase sailing speeds in order to make the vessels arrive in advance of succeeding voyages to collect rewards. Thus, the vessels might be sailing close to or on the maximum speed limit on several voyages and hence not have the ability to utilize the benefit of the ON setting.

The AST solution has the second highest and highest simulated costs with the NN and ON setting, respectively. On one hand, this may imply that it is more effective to rather reward or penalize certain characteristics in a solution, such as with the REA and PRST strategies, than restricting the solution space. On the other hand, as given in Table 14.5, the sailing time adjustment parameter Ω was only set to 2.00 %. If it was possible to increase the value of this parameter without incrementing the number of unserviced voyages, additional slack could have been added to the solution and the strategy may have performed better.

For both the NN and ON settings, the COMBINED strategy performs slightly better than all of the other robustness strategies. We see that the total delay can be reduced to as little as 77 days in total, or in average 0.347 days per voyage, by applying the COMBINED strategy when planning the fleet deployment. We see that the COMBINED strategy achieves low simulated costs for the NN setting, and yet is able to reduce the total delay from 118 to 77 days when allowing speed adjustments. It is reasonable to believe that the COMBINED strategy is able to balance the use of the REA and PRST strategies to both be able to create a robust plan that performs well with the NN setting, and a plan that is able to utilize the advantage of adjusting sailing speeds during the operation.

14.3 Evaluating the Re-planning Recovery Procedure

In this section we enable the re-planning recovery procedure in the simulation framework and examine different strategies of triggering this procedure during simulations. The functionality of the procedure is thoroughly presented in Section 13.2. In this section, we first describe the selection of an appropriate recovery strategy to be used in the re-planning recovery procedure. Then, different conditions for triggering the procedure are evaluated and discussed.

Selecting a recovery strategy

As concluded in Andersson et al. (2014) and in the computational study performed in Chapter 7, large problem instances of the FDP cannot be solved by a MIP within a reasonable amount of time. If the recovery procedure is called early in the planning horizon, only a few decisions have been made final. Now, if we use the MIP based on the mathematical formulation in Section 4.2 to solve the remaining parts of the problem, we are likely left with a time-consuming mathematical program. This is assumed to have little or no practical value for the case company. However, the RHH was found effective on larger problem instances and provided good solutions within a shorter amount of time. Therefore, we have selected the RHH as the recovery strategy to be used in the re-planning recovery procedure.

A characteristic of real-life planning that favors the use of the RHH is how the problem continuously grows. When additional future contracted voyages are confirmed, they should be included in the problem. This means that when the re-planning procedure is triggered, several additional voyages may have appeared and been added to the planning horizon. Even though the MIP may be efficient for smaller problem instances, realistic problems are large and remains, due to this characteristic, large despite fixing any previous actions.

When solving the re-planning problem with the RHH, we also include the same robustness measure(s) that were used to generate the initial solution, as described in Section 13.2. We use the following stopping criteria for the program:

1. Optimal solution.
2. Best solution after 10,000 seconds (which are equally distributed to each primary period)

Conditions for triggering the re-planning recovery procedure

When including the re-planning procedure in the program, we must determine when it should be called. As liner shipping companies have different preferences, this choice may differ from company to company. For instance, one company may accept some delays to maintain its cost-efficient plan. Other companies, however, aim to reduce delays as much as possible to keep highly demanding customers satisfied. We have tested three different trigger strategies for calling the re-planning recovery procedure. These trigger strategies are to call the procedure when:

1. at least one day delay for a voyage is anticipated with the current plan
2. at least three days delay for a voyage is anticipated with the current plan
3. at least five days delay for a voyage is anticipated with the current plan

The different trigger conditions proposed above were applied to the simulation-optimization program and tested by simulating initial plans created with the BASIC and COMBINED robustness strategies. These two strategies were chosen primarily because we wanted to examine if different trigger strategies were effective for recovering initial solutions generated with different robustness considerations.

Table 14.9 shows the results obtained by simulating BASIC and COMBINED initial solutions when the re-planning recovery procedure is activated. All the reported results are average numbers, and are due to time limitations only based on five simulations for each combination of trigger and robustness strategy. Each row contains the results for the three different trigger strategies previously proposed. The simulated costs of the solutions are reported as a percentage of the BASIC planned costs. The other three columns for each strategy report the average total delay incurred, number of times the re-planning recovery procedure is called and the number of voyage swaps performed, respectively.

Table 14.9 – Average simulated costs, total days of delay, number of times the recovery procedure is called and performed voyage swaps for different trigger conditions on BASIC and COMBINED solutions. Simulated costs of solutions are expressed as % of the BASIC planned costs.

	BASIC				**COMBINED**			
	SC[1]**(%)**	**D**[2]	**RP**[3]	**VS**[4]	**SC**[1]**(%)**	**D**[2]	**RP**[3]	**VS**[4]
1-day[5]	116.35	134	27	602	111.45	55	14	379
3-day[6]	116.11	133	11	418	111.96	60	5	293
5-day[7]	116.83	143	5	270	112.61	66	2	266

[1] Percent of planned cost.
[2] Total days of delay.
[3] Number of times the recovery re-planning procedure is called.
[4] Number of voyage swaps.
[5] Re-planning procedure triggered when a voyage is scheduled to be more than one day delayed.
[6] Re-planning procedure triggered when a voyage is scheduled to be more than three days delayed.
[7] Re-planning procedure triggered when a voyage is scheduled to be more than five days delayed.

We first observe that by including the re-planning recovery procedure in the simulation framework, significant costs savings can be achieved. For the BASIC solution, the simulated costs are, for all triggers, just above 16 % of the planned costs, compared to roughly 21 % when only speed adjustments are allowed (see Table 14.8). Cost savings are also obtained for the COMBINED solution. Second, we see that the lower threshold we use as a trigger, the more often the recovery procedure is called, as could be expected. When simulating COMBINED solutions with a 5-day voyage delay trigger, the procedure is on average only called twice. The re-planning recovery procedure is further called approximately twice as often in the BASIC simulations than in the COMBINED simulations. This illustrates that the initial COMBINED solutions are considerably more robust to disruptive events and are able to mitigate most of the impacts of the incurred disruptions with their incorporated slack.

Next, we observe how little the incurred total delay varies between the different trigger strategies. For the BASIC simulations, we observe that a 3-day trigger actually seem to perform slightly better than calling the procedure every time a voyage is anticipated to experience only one day of delay or more, both in terms of delays and costs. However, the results do not indicate the same for COMBINED solutions. We suggest that this difference is due to the recovery procedure for BASIC solutions does not have the same robustness incentives as when called on COMBINED solutions. Due to the fewer number of simulations, this should be tested further to give any final conclusions.

Naturally, the more often the recovery procedure is called, the more voyage swaps are recorded. This relationship can also be seen by comparing the number of changes of the simulated BASIC and COMBINED solutions, as both the number of changes and the number of recovery procedure calls are significantly lower for COMBINED solutions than for the BASIC solutions. A scatter plot of the BASIC and COMBINED simulated costs and number of voyage swaps resulting from the re-planning procedure with different trigger conditions is presented in Figure 14.2.

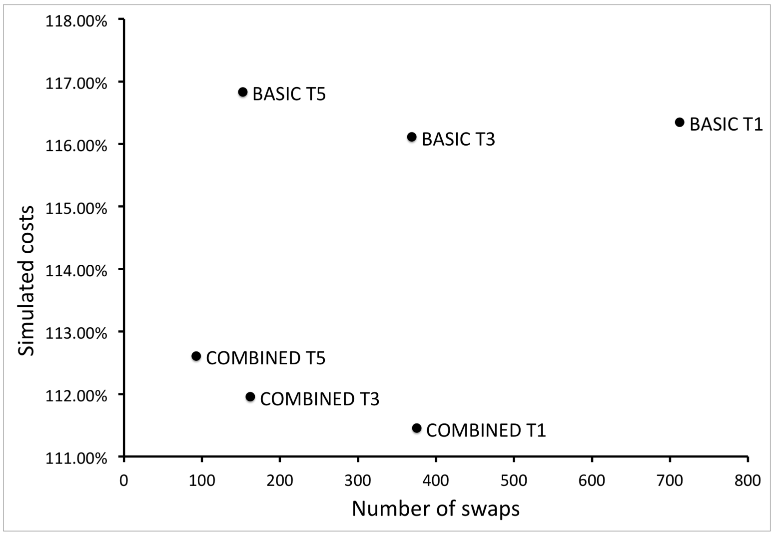

Figure 14.2 – Simulated costs and number of voyage swaps when simulating BASIC and COMBINED solutions with different trigger conditions. T1, T3 and T5 denote triggering the re-planning recovery procedure when a voyage is scheduled to be one, three or five days delayed, respectively.

By considering the aforementioned observations, we suggest that including the re-planning recovery procedure in the simulation framework has the potential of better mitigating the impacts of disruptive events than by not considering recovery actions. Despite slightly lower inflicted costs, easily triggering the re-planning recovery procedure leads to many alterations to the operations. Even though voyage swaps may only be planned changes and do not actually incur due to additional events, these are still considered as changes and must be facilitated by the case company. The results suggest that if a price for performing changes was considered, triggering the recovery procedure less often may be desirable. Different companies have their own individual preferences in the trade-off between the extent of changes, expected costs and expected delays, and therefore we do not give a final suggestion of the optimal trigger strategy.

14.4 Combining Robustness Strategies and Re-planning Recovery

In this section we present and discuss the results from simulating solutions generated by the different robustness strategies when including the re-planning recovery procedure in the simulation framework. First, we provide the numerical results from the conducted simulations. Then, a discussion of how the combination of robustness strategies and re-planning recovery performs compared to results in Section 14.2 is given. Finally, we elaborate on the trade-off between operating costs, delays and the number of changes, and follow the development of different initial solutions through a simulation.

Numerical results

Table 14.10 shows the simulated costs of the solutions obtained by the different robustness strategies combined with re-planning recovery as a percentage of the BASIC planned operating costs. For the purpose of comparing the performance of combined robustness strategies and re-planning recovery, the recovery procedure is triggered when the expected delay of any voyage exceeds three days. We also revisit the results from Table 14.8 for easier comparison.

Table 14.10 – Average simulated costs and total days of delay for the different robustness strategies combined with re-planning recovery. The simulated costs are expressed as % of the planned costs for the BASIC solution.

	BASIC		**AST**		**REA**		**PRST**		**COMBINED**	
	SC[1]**(%)**	**D**[2]	**SC**[1]**(%)**	**D**[2]	**SC**[1]**(%)**	**D**[2]	**SC**[1]**(%)**	**D**[2]	**SC**[1]**(%)**	**D**[2]
NN[3]	155.02	604	133.72	329	117.06	113	143.38	454	116.15	118
ON[3]	121.50	197	118.93	142	114.86	82	115.40	112	113.41	77
OR[4]	117.31	143	116.55	116	114.48	81	113.10	84	112.83	67

[1] Simulated costs in % of planned cost for the BASIC solution.
[2] Total days of delay.
[3] Speeds may be adjusted every day during the simulation. Other recovery actions are not allowed.
[4] Speeds are open and may be adjusted every day during the simulation. Re-planning recovery are called when the delay exceeds three days on any voyage after a disruption.

The results in Table 14.10 show that by enabling re-planning recovery in the simulations, denoted by OR, further cost and delay reductions are achieved for all robustness strategies. As for the NN and the ON settings, the BASIC solution has the poorest performance in both delay and simulated costs compared to the robust solutions. Despite the relatively high simulated costs of the BASIC solution with re-planning enabled, it is evident that including re-planning recovery could result in considerable savings. The savings obtained by including re-planning for the BASIC solution compared to only allowing speed adjustments is 4.19 %, or $9,566,853 in absolute costs for this problem instance.

Of the robustness strategies, the solution based on the AST strategy has the largest simulated costs and highest amount of delay. As suggested in Section 14.2, the poor performance of the AST solution may be a consequence of the restricted solution space or the fact that the sailing adjustment parameter value is not sufficiently high.

For the REA and PRST solutions we observe an alteration in performance. When including re-planning recovery, the PRST solution achieve a lower simulated cost than the REA solution, while the amount of days delayed for the REA solution remains the lowest between the two. The higher costs of the REA solution might be explained by the differences in the robustness measures. The REA strategy and PRST strategy both encourage the solver to add slack prior to the latest start time of a subsequent voyage. Unlike the PRST strategy, however, the REA strategy also motivates the solver to make the vessels arrive earlier than the time window of the succeeding voyage. If no disruptions occur, a vessel will in cases when it arrives before the time window of the next voyage have to wait before starting the voyage. Instead, it could have sailed at a lower speed and consequently reduced the operating costs, but still been able to sail the upcoming voyage at the planned start time. These situations will not happen for the PRST strategy, as the robustness measure only will encourage vessels to avoid starting voyages in the latest part of the time window. I.e., it will not risk to impose idle vessel time outside of time windows. Hence, when including re-planning, the results show that encouraging slacks prior to the end of time windows gives lower total costs than motivating arrivals in advance of voyage time windows.

The solution based on the COMBINED strategy results in the lowest simulated costs and number of days delayed when enabling re-planning recovery. If the case company were to implement fleet deployment plans obtained by the RHH without considering uncertainties and disruption management, and adjusted speeds on a day-to-day basis, the simulated costs would be reflected by the BASIC ON setting. Compared to the COMBINED solution with re-planning enabled, substantial savings of up to 8.67 % could be made. For the simulated problem instance this implies savings of $19,795,851 in absolute costs.

All numerical results presented in this section is based on the average of ten simulations with predefined event scenarios. A scatter plot of the average simulated costs and the average total delay for solutions based on the robustness strategies is presented in Figure 14.3. The confidence level that the simulated costs obtained with re-planning recovery for the robust solutions are lower than the simulated costs of the BASIC solution with adjustable speeds is over 99.27 %. This, together with relevant confidence levels of other comparisons in this section, are given in Appendix B.1.

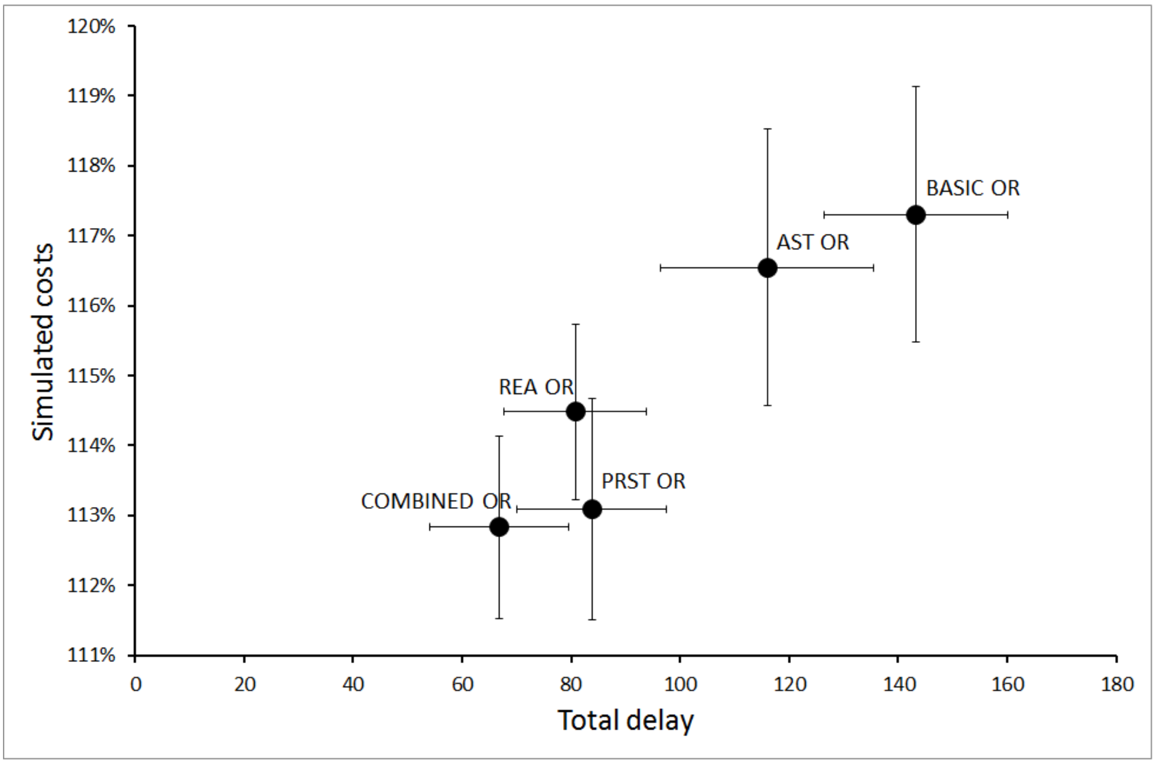

Figure 14.3 – Simulated costs and delays for each strategy with re-planning recovery enabled, with 95 % confidence intervals.

Recovery actions and extent of changes

With different robustness strategies, the number of recovery procedures called during the simulation of the planning period will vary. When the re-planning recovery procedure is called, changes to the previous plans may have been suggested. Table 14.11 shows the average number of recovery actions performed on each robust solution, and the resulting total voyage swaps from these new plans.

From Table 14.11 we can see that the plans based on the robustness strategies BASIC and AST, that gives the highest amount of delay with the ON setting, requires most

Table 14.11 – Average number of re-planning recovery procedure calls and voyage swaps when simulating solutions generated with the different robustness strategies with the re-planning recovery procedure enabled.

	BASIC	AST	REA	PRST	COMBINED
Recovery procedure calls	12	11	9	8	7
Voyage swaps	409	300	283	196	260

recovery actions to be performed during the simulation. The robustness strategies with a lower total delay will have fewer voyages with delays of over three days, and thus trigger fewer recovery actions. However, we see that the simulations of the PRST solution only trigger eight recovery procedures compared to nine for the REA solution, despite having a 112 days of delay compared to REA's 82 days of delay when simulating the solutions with adjustable speeds. This again implies that using the PRST strategy results in plans where the disruptive events will cause many smaller delays, while the REA strategy gives vessel schedules that will have fewer voyages that are delayed, but of greater magnitude.

Further, we observe that solutions generated with the PRST robustness strategy are on average subject to the least amount of voyage swaps. This can also be explained by how the PRST robustness strategy evenly distributes slack to vessels. Considering that COMBINED solutions are generated by using a combination of the robustness strategies, the amount of voyage swaps lies somewhere in between. However, the re-planning recovery procedure is only called seven times.

By comparing PRST and COMBINED, we can see that by selecting PRST and accepting operating costs that are only 0.27 % higher, considerably fewer changes are needed to be done to the deployment plan during operation. Further, if we compare the COMBINED solution with the re-planning procedure enabled to when only speed adjustments are allowed, it can be seen that zero changes to the plan can be achieved by accepting only 0.38 % higher costs. This corresponds to $867,638 in absolute for this problem instance.

Figure 14.4 illustrates how the different robustness strategies perform given the trade-off between number of changes to the plan and operating costs, with 95 % confidence intervals.

Figure 14.5 illustrates how the simulated total costs and total delay for the different robust solutions vary through the planning period when re-planning recovery is enabled. The figure is based on simulations using one of the predefined scenarios and will thus

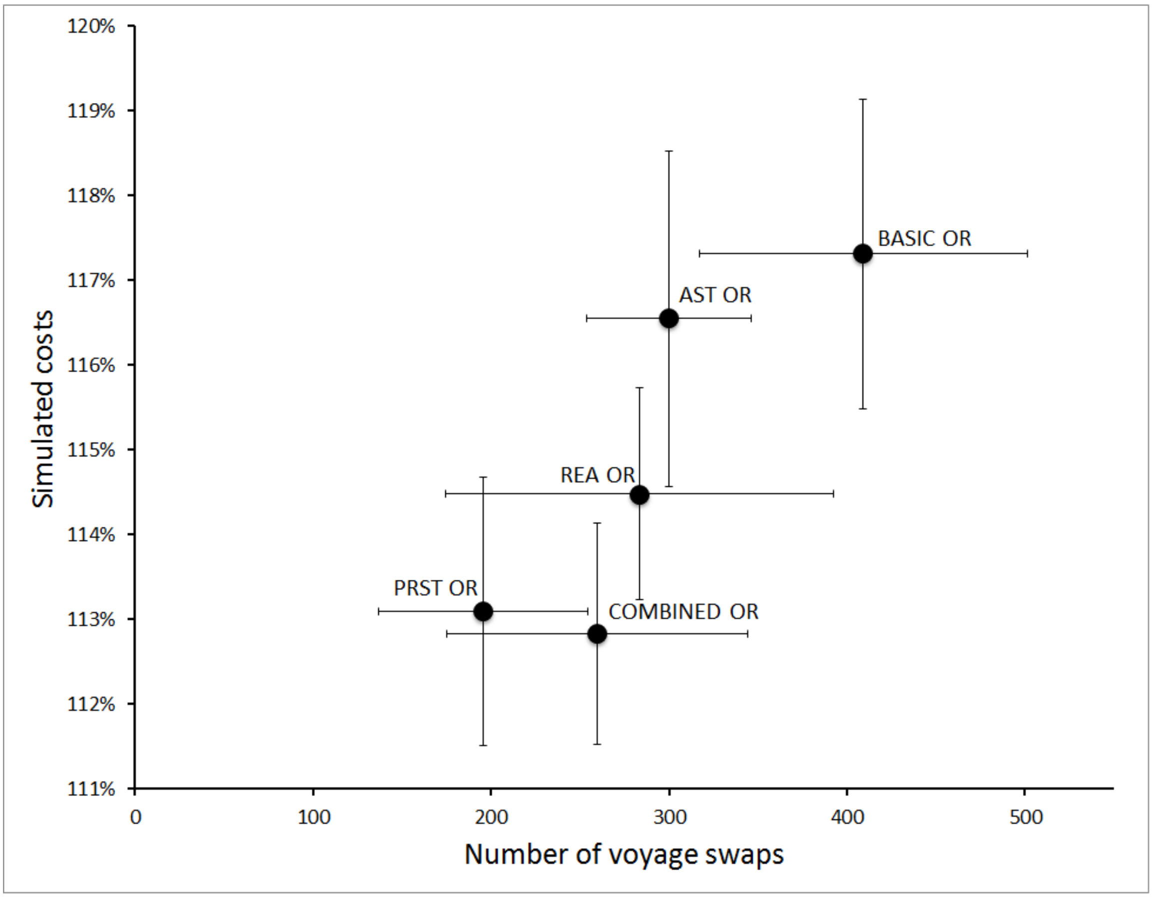

Figure 14.4 – Simulated costs and voyage swaps for each strategy with recovery re-planning enabled, with confidence 95 % intervals.

differ from the previous presented average values. The increase in total costs and delays at certain days shows the impacts of disruptions to the fleet deployment plans. The large increases in both delay and costs in the beginning of the planning period clearly display the knock-on effects of disruptions, i.e when a voyage is disrupted all the subsequent voyages are also affected. The drops in both simulated costs and delays reveal the consequences of calling re-planning recovery procedures. Also, the figure shows the large cost and delay reductions by mitigating knock-on effects when re-planning is performed in the beginning of the planning period. Further, the time between the large increases in total costs and delays, as a result of disruptions, and the drops in simulated costs and delays illustrates how the triggering mechanism for the re-planning recovery procedure works. Even though the first disruptive events lead to extensive increases in both total costs and delays, none of the voyages experience delays exceeding three days and the re-planning recovery procedure is not called. It is first when one or more voyages are expected to be more than three days delayed that re-planning is called and the simulated costs and delays are reduced. Finally, if we compare the BASIC and COMBINED solutions, we see how the COMBINED solution initially has a much larger planned cost, but the consequential costs of disruptions are significantly less and the total cost ends up much lower.

A similar comparison for solutions simulated with the ON setting is given in Appendix B.2.

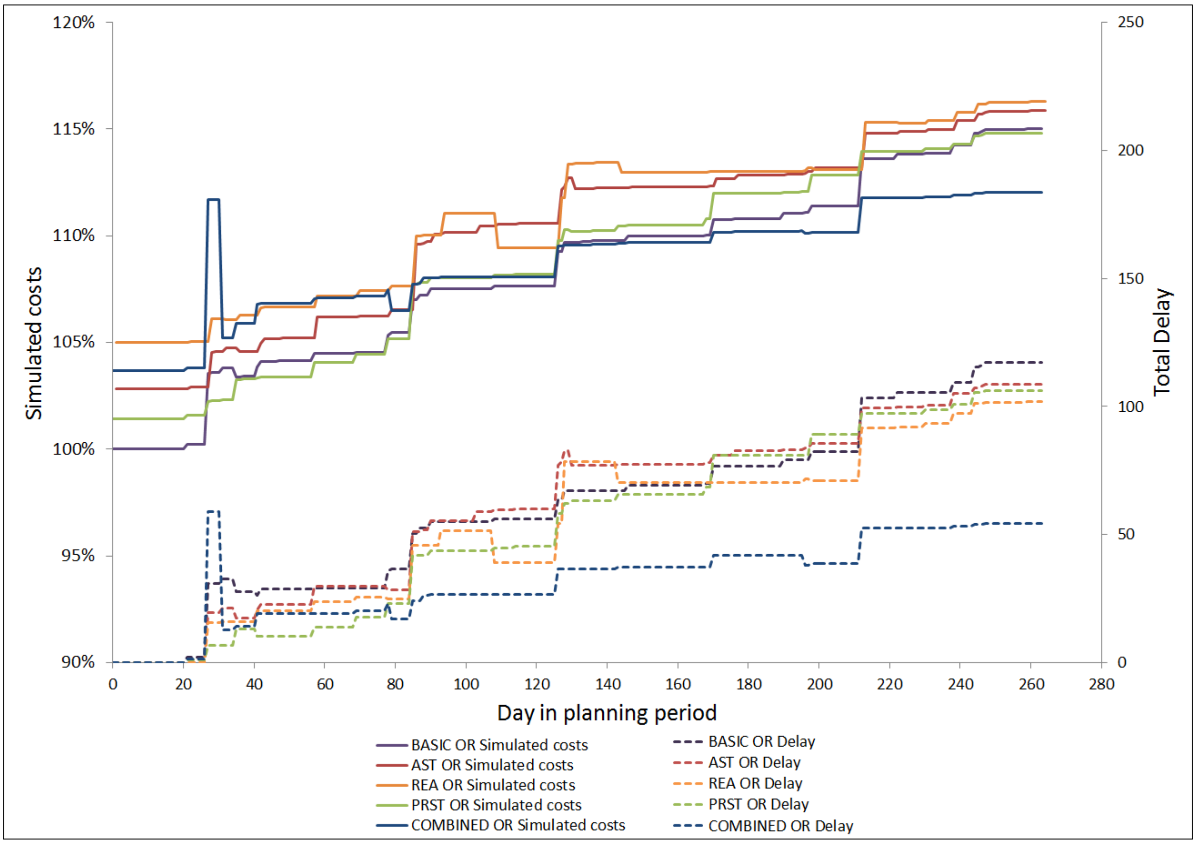

Figure 14.5 – Development of simulated costs and delay in the planning period as a result of a predetermined scenario

Chapter 15

Concluding Remarks II

In Part II of this thesis we have considered the fleet deployment problem (FDP) for a Roll-on Roll-off (RoRo) liner shipping company under uncertainty. First, we identified possible events which may affect the company's operations. Then, four different robustness strategies that can be applied when planning the fleet deployment were proposed: (1) adjust sailing times, (2) reward early arrivals, (3) penalize risky start times and (4) a combination of the three aforementioned strategies. With these four strategies we aimed to produce more robust solutions, i.e. vessel schedules less vulnerable to the impacts of disruptive events. The robustness strategies were evaluated in a developed simulation framework that randomly adds events during the planning horizon. We also presented a re-planning recovery procedure to be included in the framework. When triggered during the simulations, this procedure allows for re-planning of the remaining part of the planning horizon in order to mitigate the impacts of an occurred event. The rolling horizon heuristic found effective in Part I was selected as the recovery strategy used in the re-planning procedure.

In a computational study we simulated initial fleet deployment plans obtained with and without robustness strategies through the planning horizon. The results show that when considering uncertainties during operation, robustness measures can with great advantage be included in the planning process. We found that a combination of the suggested robustness strategies performs best overall. The presented results also indicate that including the re-planning recovery procedure may further decrease operating costs and delays, but at the cost of many voyage swaps. Compared to plans created without considering uncertainty, that was studied in Part I, robust plans and re-planning recovery may reduce incurred operating costs by up to 8.67 %. The combination robustness strategy performed nearly as good without the re-planning procedure as with, making it an excellent choice if changes during operation are undesirable. We suggest that robustness strategies and the re-planning recovery procedure based on the rolling horizon heuristic may serve as disruption management decision support tools for the case company.

Future work

During the work with Part II of this thesis, several possible future areas of research have been identified. Below we suggest what parts of our work that require further testing, how the evaluation framework can be made more realistic and how RoRo liner shipping companies can incorporate the presented work as decision support tools for disruption management.

First, due to the time limitations of this work not all aspects of the presented robustness strategies were subject to testing. A thorough review may find more effective parameter values for the given robustness strategies, or reveal additional robustness measures effective in creating robust solutions to the FDP. For instance, a plan based on the REA strategy will with our parameters and formulation encourage vessels to arrive at ports up to two days before the the start of the time window for the next voyage. To create this buffer time, extra costs related to higher speeds when sailing will be added to the operating costs. However, if no disruptive events impose delay on the vessel schedules, the two days, originally intended as buffer time for the vessels to get back on schedule in case of disruptions, will not be used. Finding other ways to reward early arrivals is therefore suggested as possible future work. In addition, time limitations restricted the number of simulations conducted in the computational study. More simulations could have improved the confidence of our results.

The simulation framework used in the computational study can, with more input from the case company and with results from further testing, be adjusted to be more realistic in terms of event probabilities and impacts. This may, e.g., highlight if the probabilities of disruptive events should be increased on certain routes during certain time periods of the year, such as during the typhoon season in Asia and the hurricane season in North America. The uncertainty in demand and transported cargo may, if further research reveals this as important, also be added to the simulation framework.

Finally, given the findings of this thesis, the case company should consider incorporating robustness strategies in fleet deployment planning to potentially mitigate effects of disruptive events during operations. Before implementing, further studies in closer collaboration with the case company should be conducted. These studies may reveal company preferences regarding number of changes, delays and expected costs, and decision support and disruption management tools can be tailored to the case company's needs.

Bibliography

Alvarez, J. F.; Tsilingiris, P.; Engebrethsen, E. S., and Kakalis, N. M. P. Robust fleet sizing and deployment for industrial and independent bulk ocean shipping companies. *INFOR: Information Systems and Operational Research*, 49(2):93–107, 2011.

Andersen, M. W. *Service Network Design and Management in Liner Container Shipping Applications*. PhD thesis, Technical University of Denmark, Department of Transport, 2010.

Andersson, H.; Fagerholt, K., and Hobbesland, K. Integrated maritime fleet deployment and speed optimization: Case study from roro shipping. *Computers & Operations Research*, 55:233–240, 2014.

Baker, K. R. An experimental study of the effectiveness of rolling schedules in production planning. *Decision Sciences*, 8(1):19–27, 1977.

Baker, K. R. and Peterson, D. W. An analytic framework for evaluating rolling schedules. *Management Science*, 25(4):341–351, 1979.

Barnhart, C. Irregular operations: Schedule recovery and robustness. *The Global Airline Industry*, pages 253–274, 2009.

BBC. Crew rescued from listing solent car transporter, 2015. URL `http://www.bbc.com/news/uk-england-hampshire-30670746`. Accessed: 2015-05-19.

Bertsimas, D. and Sim, M. Robust discrete optimization and network flows. *Mathematical programming*, 98(1-3):49–71, 2003.

Bredström, D. and Rönnqvist, M. Supply chain optimization in pulp distribution using a rolling horizon solution approach. *NHH Dept. of Finance & Management Science Discussion Paper*, (2006/17), 2006.

Bredström, D.; Flisberg, P., and Rönnqvist, M. A new method for robustness in rolling horizon planning. *International Journal of Production Economics*, 143(1):41–52, 2013.

Brouer, B. D.; Dirksen, J.; Pisinger, D.; Plum, C. E. M., and Vaaben, B. The Vessel Schedule Recovery Problem (VSRP) – a MIP model for handling disruptions in liner shipping. *European Journal of Operational Research*, 224(2):362 – 374, 2013.

Christiansen, M. and Fagerholt, K. Robust ship scheduling with multiple time windows. *Naval Research Logistics (NRL)*, 49(6):611–625, 2002.

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

Christiansen, M.; Fagerholt, K.; Nygreen, B., and Ronen, D. Maritime transportation. *Transportation*, 14:189–284, 2007.

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

Clausen, J.; Larsen, J.; Larsen, A., and Hansen, J. Disruption management-operations research between planning and execution. Technical report, 2001.

Clausen, J.; Larsen, A.; Larsen, J., and Rezanova, N. J. Disruption management in the airline industry—concepts, models and methods. *Computers Operations Research*, 37 (5):809 – 821, 2010.

Dellaert, N. and Jeunet, J. Controlling multi-level production in a rolling-schedule environment. *International Journal of Production Economics*, 85(1):113 – 121, 2003.

Dienst, D.; Røpke, S., and Vaaben, B. Realistic models and computational results for disruption management in the airline industry. Technical report, Technical University of Denmark, 2012.

Dirksen, J. Disruption management in liner shipping: Introducing the vessel schedule recovery problem. Master's thesis, Technical University of Denmark, 2011.

Ehrgott, M. *Multicriteria optimization*. Springer Science & Business Media, 2006.

Ehrgott, M. and Ryan, D. M. Constructing robust crew schedules with bicriteria optimization. *Journal of Multicriteria Decision Analysis*, 11(3):139, May 2002.

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

Fischer, A. S.; Nokhart, H. S., and Olsen, H. A. The fleet deployment problem in RoRo shipping. Project Thesis TIØ4500, Fall, 2014.

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

GettyImages. Höegh Osaka tilted. Photographer: Peter Macdiarmid. Accessed: 2015-02-01.

Grand Port Maritime du Havre. Ro-Ro terminal facilities in Le Havre, France. `http://www.terminal-roro-lehavre.fr/en/roro-terminal/ro-ro-terminal-facilities-le-havre`. Accessed: 2015-05-08.

Gurobi Optimization, I. Gurobi optimizer reference manual, 2014. URL `http://www.gurobi.com`.

Halvorsen-Weare, E. E. and Fagerholt, K. Robust supply vessel planning. In *Network Optimization*, pages 559–573. Springer, 2011.

Halvorsen-Weare, E. E.; Fagerholt, K.; Nonås, L. M., and Asbjørnslett, B. E. Optimal fleet composition and periodic routing of offshore supply vessels. *European Journal of Operational Research*, 223(2):508–517, 2012.

Halvorsen-Weare, E. E.; Fagerholt, K., and Rönnqvist, M. Vessel routing and scheduling under uncertainty in the liquefied natural gas business. *Computers & Industrial Engineering*, 64(1):290–301, 2013.

Hoff, A.; Andersson, H.; Christiansen, M.; Hasle, G., and Løkketangen, A. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061, 2010.

Hua, E. Growing pains at china's top ro-ro ports, 2013. URL `http://www.automotivelogisticsmagazine.com/intelligence/navigating-growth-at-chinas-top-ro-ro-ports`. Accessed: 2015-04-22.

International Cargo/Export Shipping Co., 2014. URL `http://www.exship.com/images/Shipillustrated.jpg`. Accessed: 2014-10-11.

Jaramillo, D. I. and Perakis, A. N. Fleet deployment optimization for liner shipping part 2. implementation and results. *Maritime Policy and Management*, 18(4):235–262, 1991.

Kauczynski, W. Study of the reliability of the ship transportation. In *Proceeding of the international conference on ship and marine research*, page 15, 1994.

Kjeldsen, K. H.; Ergun, O.; Lysgaard, J., and Erera, A. Rescheduling ships and cargo in liner shipping in the event of disruptions. *Liner Shipping*, page 105, 2011.

Korsvik, J. E.; Fagerholt, K., and Laporte, G. A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, 38 (2):474–483, 2011.

Kouvelis, P. and Yu, G. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 1997.

Lawrence, S. A. *International sea transport: the years ahead.* Lexington Books Lexington, MA, 1972.

Lindstad, H.; Asbjørnslett, B. E., and Strømman, A. H. The importance of economies of scale for reductions in greenhouse gas emissions from shipping. *Energy Policy*, 46: 386–398, 2012.

List, G. F.; Wood, B.; Nozick, L. K.; Turnquist, M. A.; Jones, D. A.; Kjeldgaard, E. A., and Lawton, C. R. Robust optimization for fleet planning under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 39(3):209 – 227, 2003.

Liu, X.; Ye, H.-Q., and Yuan, X.-M. Tactical planning models for managing container flow and ship deployment. *Maritime Policy & Management*, 38(5):487–508, 2011.

Lundgren, J.; Rönnqvist, M., and Värbrand, P. *Optimization.* Studentlitteratur AB, 2010.

Meng, Q. and Wang, T. A chance constrained programming model for short-term liner ship fleet planning problems. *Maritime Policy & Management*, 37(4):329–346, 2010.

Mercé, C. and Fontan, G. MIP-based heuristics for capacitated lotsizing problems. *International Journal of Production Economics*, 85(1):97–111, 2003.

NBC News. Ports Workers Face Huge Backlog After Dispute Ends, 2015. URL `http://www.nbcnews.com/business/business-news/`

ports-workers-face-huge-backlog-after-dispute-ends-n310346. Accessed: 2015-05-12.

Norstad, I.; Fagerholt, K., and Laporte, G. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19(5): 853 – 865, 2011.

Notteboom, T. E. The time factor in liner shipping services. *Maritime Economics & Logistics*, 8(1):19–39, 2006.

Notteboom, T. E. and Rodrigue, J.-P. Containerisation, box logistics and global supply chains: The integration of ports and liner shipping networks. *Maritime Economics & Logistics*, 10(1):152–174, 2008.

Nygreen, B.; Christiansen, M.; Haugen, K.; Bjørkvoll, T., and Kristiansen, Ø. Modeling norwegian petroleum production and transportation. *Annals of Operations Research*, 82:251–268, 1998.

Øvstebø, B. O.; Hvattum, L. M., and Fagerholt, K. Optimization of stowage plans for RoRo ships. *Computers & Operations Research*, 38(10):1425–1434, 2011.

Paul, J. A. and Maloni, M. J. Modeling the effects of port disasters. *Maritime Economics & Logistics*, 12(2):127–146, 2010.

Perakis, A. N. and Jaramillo, D. I. Fleet deployment optimization for liner shipping part 1. Background, problem formulation and solution approaches. *Maritime Policy & Management*, 18(3):183–200, 1991.

Pisinger, D. and Ropke, S. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

Pisinger, D. and Ropke, S. Large neighborhood search. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 39–419. Springer US, 2010.

Powell, B. J. and Perakis, A. N. Fleet deployment optimization for liner shipping: An integer programming model. *Maritime Policy & Management*, 24(2):183–192, 1997.

Qi, X. Disruption management for liner shipping. In Lee, C.-Y. and Meng, Q., editors, *Handbook of Ocean Container Transport Logistics*, volume 220 of *International Series in Operations Research Management Science*, pages 231–249. Springer International Publishing, 2015.

Rakke, J. G.; Stålhane, M.; Moe, C. R.; Christiansen, M.; Andersson, H.; Fagerholt, K., and Norstad, I. A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. *Transportation Research Part C: Emerging Technologies*, 19 (5):896–911, 2011.

Ribeiro, G. M. and Laporte, G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728–735, 2012.

Rodrigue, J.-P.; Comtois, C., and Slack, B. *The geography of transport systems*. Routledge, 2013.

Ronen, D. Cargo ships routing and scheduling: Survey of models and problems. *European Journal of Operational Research*, 12(2):119–126, 1983.

Ropke, S. and Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

Shaw, P. A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, 1997.

Sherali, H. D.; Al-Yakoob, S. M., and Hassan, M. M. Fleet management models and algorithms for an oil-tanker routing and scheduling problem. *IIE Transactions (Institute of Industrial Engineers)*, 31(5):395–406, 1999.

Sinclair, K.; Cordeau, J.-F., and Laporte, G. Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, 233(1):234–245, 2014.

Sitompul, C. and Aghezzaf, E.-H. Robust production planning: An alternative to scenario-based optimization models. In *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pages 328–337. Springer, 2008.

Smith, B. C. and Johnson, E. L. Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science*, 40(4):497–516, 2006.

Sokol, J.; Zhang, C.; Nemhauser, G.; Papageorgiou, D., and Cheon, M.-S. Robust inventory routing with flexible time window allocation. *Working paper*, 2015.

Stauffer, L. and Liebling, T. M. Rolling horizon scheduling in a rolling-mill. *Annals of Operations Research*, 69(0):323–349, 1997.

Stopford, M. Maritime economics, 3rd, 2009.

Stålhane, M.; Rakke, J. G.; Moe, C. R.; Andersson, H.; Christiansen, M., and Fagerholt, K. A construction and improvement heuristic for a liquefied natural gas inventory routing problem. *Computers & Industrial Engineering*, 62(1):245 – 255, 2012.

Teodorović, D. and Guberinić, S. Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, 15(2):178–182, 1984.

Tierney, K. and Jensen, R. M. The liner shipping fleet repositioning problem with cargo flows. In *Computational Logistics*, pages 1–16. Springer, 2012.

Tierney, K.; Áskelsdóttir, B.; Jensen, R. M., and Pisinger, D. Solving the liner shipping fleet repositioning problem with cargo flows. *Transportation Science*, 2014.

UNCTAD. Review of maritime transport, 2013. URL `http://unctad.org/en/publicationslibrary/rmt2013_en.pdf`. United Nations Conference on Trade and Development. UNCTAD/RMT/2013. Sales no. E.13.II.D.9, United Nations publication, 2013.

Vernimmen, B.; Dullaert, W., and Engelen, S. Schedule unreliability in liner shipping: Origins and consequences for the hinterland supply chain. *Maritime Economics and Logistics*, 9(3):193–213, 2007.

VR WORLD. Vessels waiting outside West Coast ports during the strike in February, 2015. `http://www.vrworld.com/2015/02/13/west-coast-port-strike-wreaking-havoc-pc-industry/`. Accessed: 2015-05-12.

Wang, S. and Meng, Q. Liner ship fleet deployment with container transshipment operations. *Transportation Research Part E: Logistics and Transportation Review*, 48(2): 470–484, 2012.

Wang, T.; Meng, Q.; Wang, S., and Tan, Z. Risk management in liner ship fleet deployment: A joint chance constrained programming model. *Transportation Research Part E: Logistics and Transportation Review*, 60:1–12, 2013.

Wiki Commons. Höegh Osaka, 2015. URL `http://www.tu.no/industri/2015/01/05/overingenior-dette-kan-vare-arsaken-til-slagsiden`. Accessed: 2015-06-05.

World Shipping Council. Valuation of the liner shipping industry, 2009. URL http://www.worldshipping.org/pdf/Liner_Industry_Valuation_Study.pdf.

WWLBreakbulk, 2010. URL https://www.youtube.com/watch?v=dDn7y4iTRTQ. Screenshot at 2:05. Accessed: 2014-10-03.

Yu, G. and Qi, X. *Disruption Management: Framework, Models and Applications*. World Scientific, 2004.

Zeng, Q.; Yang, Z., and Chen, C. Robust optimization model for resource allocation of container shipping lines. *Tsinghua Science & Technology*, 15(5):586–594, 2010.

Appendix A

A.1 Mathematical Model

In this section, a compact version of the mathematical formulation presented in Chapter 4 is provided.

Summary of Notation

Sets

$\mathcal{V}$	-	Vessels, indexed by v
$\mathcal{S}_v$	-	Discrete speed options for vessel v, indexed by s
$\mathcal{R}$	-	Standard routes, indexed by r
$\mathcal{P}$	-	Cargo segments, indexed by p
$\mathcal{P}_b$	-	Set of all products associated with b, indexed by p
$\mathcal{B}$	-	Balance categories, indexed by b
$\mathcal{B}_r$	-	Balance categories serviced by r, indexed by b
$\mathcal{B}_\xi$	-	Balance categories in capacity group ξ, indexed by b
$\mathcal{M}$	-	Set of months, indexed by m
$\mathcal{K}$	-	Set of capacity classes, indexed by k
$\mathcal{K}_p$	-	Set of capacity classes that can carry cargo segment p, indexed by k
Ξ	-	Set of capacity groups, indexed by ξ
$\mathcal{N}$	-	Set of nodes, indexed by (r, i)
$\mathcal{N}^C$	-	Set of nodes that are contracted voyages, indexed by (r, i)
$\mathcal{N}_v$	-	Set of nodes that v can service, indexed by (r, i)
$\mathcal{N}_v^M$	-	Set of nodes corresponding to required dock maintenance visits for vessel v, indexed by (r, i)

$\mathcal{I}_{rm}$	-	Set of voyages on trade route r in month m, indexed by i
$\mathcal{A}$	-	Set of arcs, indexed by $((r,i),(q,j))$
$\mathcal{A}_v$	-	Set of arcs that ship v can sail, indexed by $((r,i),(q,j))$

Parameters

D_{bpm}	-	Demand in month m for product (b,p)
$\overline{Q}^{SC}_{bpm}$	-	Maximum allowed space chartering volume of product (p,b) in month m
Q_{vrk}	-	Default capacity for vessel v on route r per capacity class k
Q^{MAX}_k	-	Maximum capacity of capacity class k in the fleet
Q^S_k	-	Capacity for the spot vessels per capacity class k
T_{vris}	-	The time it takes ship v to sail voyage (r,i) at speed s
T^B_{vriqjs}	-	The time it takes ship v to sail ballast between voyage (r,i) and (q,j) at speed s
$T^B_{o(v)ris}$	-	The time of vessel v sailing ballast from its origin to the start of voyage (r,i) at speed s
C_{vris}	-	The cost of vessel v sailing voyage (r,i) at speed s
C^B_{vriqjs}	-	The cost of vessel v sailing ballast from the end of voyage (r,i) to the start of voyage (q,j) at speed s
$C^B_{o(v)ris}$	-	The cost of vessel v sailing ballast from its origin to the start of voyage (r,i) at speed s
$C_{o(v)d(v)}$	-	The cost of vessel v sailing directly from its origin to its destination, i.e. not using vessel v for any voyages in the planning horizon
C^{SC}_{bpm}	-	Cost per unit shipped with space chartering
C^P	-	Penalty cost per day the start of a voyage is delayed
C^S	-	The cost of not servicing a voyage
$E_{o(v)}$	-	Earliest availability of ship v
E_{ri}	-	Time window start of voyage (r,i)
L_{ri}	-	Time window end of voyage (r,i)
D^{MAX}	-	The maximum days of delay allowed for starting a voyage after L_{ri}

Variables

$y_{o(v)ri}$	-	1 if vessel v sails directly from its initial position $o(v)$ to the starting point of voyage (r, i), 0 otherwise
y_{vri}	-	1 if vessel v sails voyage (r, i), 0 otherwise
y_{vriqj}	-	1 if vessel v sails ballast from voyage (r, i) to voyage (q, j), 0 otherwise
$y_{rid(v)}$	-	1 if vessel v sails voyage (r, i) as the last voyage, 0 otherwise
$y_{o(v)d(v)}$	-	1 if vessel v sails directly from its origin to the artificial destination and is not used during the planning period
y^S_{ri}	-	1 if the voyage (r, i) is not serviced
x_{vris}	-	the weight of speed alternative s for sailing voyage (r, i) by vessel v
$x^B_{o(v)ris}$	-	the weight of speed alternative s for the sailing from the initial position $o(v)$ of ship v to the first port of voyage (r, i)
x^B_{vriqjs}	-	the weight of speed alternative s for sailing ballast between voyage (r, i) and voyage (q, j) by ship v
$t_{o(v)}$	-	the starting time for vessel v from initial position $o(v)$
t_{ri}	-	the start time of voyage (r, i)
d_{ri}	-	number of days voyage (r, i) is delayed
l_{ribpk}	-	the loaded volume of product (b, p) on capacity class k on voyage (r, i)
z_{bpm}	-	the volume of product (b, p) covered with space charter in month m

Objective function

$$\begin{aligned}\min \text{ z} = &\sum_{v\in\mathcal{V}}\sum_{(r,i)\in\mathcal{N}_v}\sum_{s\in\mathcal{S}_v} C_{vris}x_{vris} + \sum_{v\in\mathcal{V}}\sum_{((r,i),(q,j))\in\mathcal{A}_v}\sum_{s\in\mathcal{S}_v} C^B_{vriqjs}x^B_{vriqjs} + \\ &\sum_{v\in\mathcal{V}}\sum_{(r,i)\in\mathcal{N}_v}\sum_{s\in\mathcal{S}_v} C^B_{o(v)ris}x^B_{o(v)ris} + \sum_{v\in\mathcal{V}} C_{o(v)d(v)}y_{o(v)d(v)} + \\ &\sum_{(r,i)\in\mathcal{N}} C^P d_{ri} + \sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}} C^{SC}_{bpm}z_{bpm} + \sum_{(r,i)\in\mathcal{N}} C^S y^S_{ri}\end{aligned} \tag{A.1}$$

The objective function (A.1) minimizes costs related to operating the fleet of vessels and fulfilling the constraints below. The cost of sailing voyages and ballast between voyages with the speed profile determined by x_{vris} and x^B_{vriqjs} is included in the two first expressions. The third expression addresses costs for the ballast sailing between origin and the first voyage for each vessel, and the cost of not using a vessel in the planning period included in the fourth expression. The last line includes the costs for delay in servicing voyage (r, i), costs related to chartering space and the cost for not servicing voyage (r, i) by the company's fleet.

Constraints

$$y_{o(v)ri} - \sum_{s \in \mathcal{S}_v} x^B_{o(v)ris} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v, \tag{A.2}$$

$$y_{vri} - \sum_{s \in \mathcal{S}_v} x_{vris} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v, \tag{A.3}$$

$$y_{vriqj} - \sum_{s \in \mathcal{S}_v} x^B_{vriqjs} = 0, \qquad v \in \mathcal{V}, ((r, i), (q, j)) \in \mathcal{A}_v. \tag{A.4}$$

Constraints (A.2)-(A.4) connect the speed and flow variables.

$$\sum_{v \in \mathcal{V}} y_{vri} + y^S_{ri} = 1, \qquad (r, i) \in \mathcal{N}^{\mathcal{C}}. \tag{A.5}$$

Constraints (A.5) ensure that all contracted voyages are served.

$$y_{vri} = 1, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}^M_v. \tag{A.6}$$

Constraints (A.6) make sure that vessel v visits a maintenance node if the set $\mathcal{N}^M_v$ is non-empty.

$$\sum_{(r,i) \in \mathcal{N}_v} y_{o(v)ri} = 1 - y_{o(v)d(v)}, \qquad v \in \mathcal{V}, \tag{A.7}$$

$$\sum_{(r,i) \in \mathcal{N}_v} y_{rid(v)} = 1 - y_{o(v)d(v)}, \qquad v \in \mathcal{V}, \tag{A.8}$$

$$y_{vri} - y_{o(v)ri} - \sum_{(q,j) \in \mathcal{N}_v} y_{vqjri} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v, \tag{A.9}$$

$$y_{vri} - y_{rid(v)} - \sum_{(q,j) \in \mathcal{N}_v} y_{vriqj} = 0, \qquad v \in \mathcal{V}, (r, i) \in \mathcal{N}_v. \tag{A.10}$$

Constraints (A.7)-(A.10) are flow variables for each vessel.

$$y_{o(v)ri}(t_{o(v)} - t_{ri} + \sum_{s \in \mathcal{S}_v} T^B_{o(v)ris} x^B_{o(v)ris}) \leq 0, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v, \tag{A.11}$$

$$y_{vriqj}(t_{ri} - t_{qj} + \sum_{s \in \mathcal{S}_v} (T^B_{vriqjs} x^B_{vriqjs} + T_{vris} x_{vris}) \leq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v. \tag{A.12}$$

Constraints (A.11) make sure that the starting time of the next voyage is greater than or equal to the starting time of the initial position plus the time spent ballast sailing from this position, while constraints (A.12) make sure that the starting of the next voyage is greater than or equal to the starting time of the previous voyage plus the time spent sailing it and the ballast sailing between the voyages. In order to implement these constraints in a commercial optimization solver, they need to be linearized. This is shown in Appendix A.2.

$$d_{ri} \geq t_{ri} - L_{ri}, \qquad (r,i) \in \mathcal{N}. \tag{A.13}$$

Constraints (A.13) count the number of days a voyage is delayed.

$$\sum_{r \in \mathcal{R}} \sum_{i \in \mathcal{I}_{rm}} \sum_{k \in \mathcal{K}} l_{ribpk} + z_{bpm} = D_{bpm}, \qquad m \in \mathcal{M}, b \in \mathcal{B}, p \in \mathcal{P}_b. \tag{A.14}$$

Constraints (A.14) ensure that the volume transported or space chartered is equal to the monthly demand.

$$z_{bpm} \leq \overline{Q}^{SC}_{bpm}, \qquad b \in \mathcal{B}, p \in \mathcal{P}_b, m \in \mathcal{M}. \tag{A.15}$$

Constraints (A.15) limit the space chartered volume to be lower than or equal to the maximum monthly volume allowed.

$$\sum_{b \in \mathcal{B}_\xi} \sum_{p \in \mathcal{P}_b} l_{ribpk} - \left(\sum_{v \in \mathcal{V}} Q_{vrk} y_{vri} + Q^S_k y^S_{ri} \right) \leq 0, \qquad (r,i) \in \mathcal{N}, k \in \mathcal{K}_p, \xi \in \Xi. \tag{A.16}$$

Constraints (A.16) limits the total flow of cargoes in a capacity group on a voyage to be within the capacity of the vessel sailing it. This vessel could either be one from the case company's fleet or a spot vessel.

$$y_{o(v)ri} \in \{0,1\}, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v, \tag{A.17}$$

$$y_{vri} \in \{0,1\}, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v, \tag{A.18}$$

$$y_{vriqj} \in \{0,1\}, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v, s \in \mathcal{S}_v, \tag{A.19}$$

$$y_{vrid(v)} \in \{0,1\}, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v, \tag{A.20}$$

$$y_{o(v)d(v)} \in \{0,1\}, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v, s \in \mathcal{S}_v, \tag{A.21}$$

$$y^S_{ri} \in \{0,1\}, \qquad (r,i) \in \mathcal{N}_v, \tag{A.22}$$

$$x^B_{o(v)ris} \geq 0, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v, s \in \mathcal{S}_v, \tag{A.23}$$

$$x_{vris} \geq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v, s \in \mathcal{S}_v, \tag{A.24}$$

$$x^B_{vriqjs} \geq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v, s \in \mathcal{S}_v, \tag{A.25}$$

$$t_{o(v)} \geq E_{o(v)}, \qquad v \in \mathcal{V}, \tag{A.26}$$

$$t_{ri} \geq E_{ri}, \qquad (r,i) \in \mathcal{N}, \tag{A.27}$$

$$l_{ribpk} \geq 0, \qquad (r,i) \in \mathcal{N}_v, b \in \mathcal{B}_r, p \in \mathcal{P}_b, k \in \mathcal{K}, \tag{A.28}$$

$$z_{bpm} \geq 0, \qquad b \in \mathcal{B}, p \in \mathcal{P}, m \in \mathcal{M}, \tag{A.29}$$

$$0 \leq d_{ri} \leq D^{MAX}, \qquad (r,i) \in \mathcal{N}. \tag{A.30}$$

Constraints (A.17)-(A.30) are constraints defining the variables.

A.2 Linearizing the Time Constraints

In section (4.1), we presented the time constraints (4.11) and (4.12). Constraints (4.11) make sure that the starting time of the next voyage is greater than or equal to the starting time of the initial position plus the time spent ballast sailing from this position, while constraints (4.12) make sure that the starting of the next voyage is greater than or equal to the starting time of the previous voyage plus the time spent actually sailing it and following the ballast sailing between the voyages. For the purposes of this section, let us label the constraints (4.11) and (4.12) for (A.31) and (A.32):

$$y_{o(v)ri}(t_{o(v)} - t_{ri} + \sum_{s\in\mathcal{S}_v} T^B_{o(v)ris} x^B_{o(v)ris}) \leq 0, \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v \tag{A.31}$$

$$y_{vriqj}(t_{ri} - t_{qj} + \sum_{s\in\mathcal{S}_v} (T_{vris}x_{vris} + T^B_{vriqjs} x^B_{vriqjs})) \leq 0, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \tag{A.32}$$

In order to implement these constraints into commercial optimization software, they need to be linearized. This is done by using the big-M method. From this, constraints (A.33) and (A.34) are obtained:

$$t_{ri} \geq t_{o(v)} + \sum_{s\in\mathcal{S}_v} T^B_{o(v)ris} x^B_{o(v)ris} - M_1(1 - y_{o(v)ri}), \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v \tag{A.33}$$

$$t_{qj} \geq t_{ri} + \sum_{s\in\mathcal{S}_v} T^B_{vriqjs} x^B_{vriqjs} + \sum_{s\in\mathcal{S}_v} T_{vris}x_{vris} - M_2(1 - y_{vriqj}), \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \tag{A.34}$$

The necessary values for M_1 and M_2 to make these constraints behave as intended are calculated in the next subsection.

A.3 Calculating Sufficiently Large Big-M's

In this section we calculate sufficiently large values for the big-M's used in the linearized time constraints (A.33) and (A.34), and in the constraints used for creating robust solutions (12.6) and (12.7).

The Big-M's in the time constraints

Let us first calculate M_1 and M_2. The constraints (A.33) and (A.34) are revisited below:

$$t_{ri} \geq t_{o(v)} + \sum_{s\in\mathcal{S}_v} T^B_{o(v)ris} x^B_{o(v)ris} - M_1(1 - y_{o(v)ri}), \qquad v \in \mathcal{V}, (r,i) \in \mathcal{N}_v \quad \text{(A.33 revisited)}$$

$$t_{qj} \geq t_{ri} + \sum_{s \in \mathcal{S}_v} T^B_{vriqjs} x^B_{vriqjs} + \sum_{s \in \mathcal{S}_v} T_{vris} x_{vris} - M_2(1 - y_{vriqj}), \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v$$

(A.34 revisited)

Let us first consider constraints (A.33). When $y_{o(v)ri} = 1$, the magnitude of M_1 does not matter as it will be multiplied by zero. This means that vessel v sails voyage (r,i) directly from its initial position. The remaining expression describes how the starting time of voyage (r,i) must be greater than or equal to the starting time from the initial position plus the time spent sailing to the first voyage. When $y_{o(v)ri} = 0$, all associated $x^B_{o(v)ris}$ are also equal to 0, and the constraint should be redundant. Because the greatest possible value of $t_{o(v)}$ and t_{ri} is the duration of the planning horizon plus any delay allowed, we can set

$$M_1 = \text{total days in the planning horizon} + \text{days of delay allowed}. \tag{A.35}$$

Determining M_2 in constraints (A.34) becomes more complex due to the combination of the variables x^B_{vriqjs} and x_{vris}. When y_{vriqj} is 1, both of the speed variables will have values, and the start time of voyage (q,j) has to be greater than the start time of voyage (r,i) plus the time it takes to sail the voyage and the ballast between. However, when y_{vriqj} is zero, only x^B_{vriqj} has to be zero. Even if the vessel is not sailing directly to (q,j), the vessel can still sail voyage (r,i) and x_{vris} will thus have values. To prevent the constraints from being restrictive for a voyage not being sailed by the vessel, M_2 is set to the sum of the duration of the planning period, the number of days allowed of delay, and the duration of the longest voyage.

$$\begin{aligned} M_2 = {} & \text{total days in the planning horizon} + \text{days of delay allowed} \\ & + \text{duration of longest voyage}. \end{aligned} \tag{A.36}$$

The Big-M's in the reward constraints

In this section we calculate the values of M^{R1}-M^{R4} found in constraints (12.6) and (12.7), which are used to model the reward of early arrivals in Section 12.3. The linearized counterparts of the constraints are revisited below.

$$M_3^R(1-\delta_{qj}^R) \geq t_{qj}^R + t_{ri} + \sum_{s\in\mathcal{S}_v}(T_{vriqjs}^B x_{vriqjs}^B + T_{vris}x_{vris})$$
$$- E_{qj}\delta_{qj}^R - M_1^R(1-y_{vriqj}), \qquad v\in\mathcal{V}, ((r,i),(q,j))\in\mathcal{A}_v. \quad \text{(12.6 revisited)}$$

$$M_4^R(1-\delta_{ri}^R) \geq t_{ri}^R + t_{o(v)} + \sum_{s\in\mathcal{S}_v} T_{o(v)ris}^B x_{o(v)ris}^B$$
$$- E_{ri}\delta_{ri}^R - M_2^R(1-y_{o(v)ri}), \qquad v\in\mathcal{V}, (r,i)\in\mathcal{N}_v. \qquad \text{(12.7 revisited)}$$

Let us first inspect constraints (12.6). Due to the two different binary variables controlling the presence of M_1^R and M_3^R, we must consider $2^2 = 4$ four possible combinations of big-M's. These combinations are:

1. $\delta_{qj}^R = 1, y_{vriqj} = 1$
2. $\delta_{qj}^R = 0, y_{vriqj} = 1$
3. $\delta_{qj}^R = 1, y_{vriqj} = 0$
4. $\delta_{qj}^R = 0, y_{vriqj} = 0$

Let us now examine the different scenarios that may arise as the binary variables take on these values:

1. When $\delta_{qj}^R = 1, y_{vriqj} = 1$, both M_1^R and M_3^R lapse as both parameters are multiplied by zero.
2. When $\delta_{qj}^R = 0, y_{vriqj} = 1$, M_1^R is insignificant due to being multiplied with zero. The remaining of the constraints are then

$$M_3^R \geq t_{ri} + \sum_{s\in\mathcal{S}_v}(T_{vriqjs}^B x_{vriqjs}^B + T_{vris}x_{vris}), \qquad v\in\mathcal{V}, ((r,i),(q,j))\in\mathcal{A}_v.$$

From this we see that

$$M_3^R \geq \text{total days in the planning horizon}$$
$$+ \text{days of delay allowed} + \text{duration of longest voyage.}$$

3. When $\delta_{qj}^R = 1, y_{vriqj} = 0$, M_3^R is insignificant due to being multiplied with zero. The remaining of then constraints are then

$$M_1^R \geq t_{qj}^R + t_{ri} + \sum_{s \in \mathcal{S}_v} T_{vris} x_{vris} - E_{qj} \delta_{qj}^R, \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v. \tag{12.6 revisited}$$

   Similarly as in (2), from this we see that

$$\begin{aligned} M_1^R \geq & \text{ total days in the planning horizon} \\ & + \text{days of delay allowed} + \text{duration of longest voyage}. \end{aligned}$$

4. When $\delta_{qj}^R = 0, y_{vriqj} = 0$, we are left with the equations

$$M_3^R + M_1^R \geq t_{ri} + \sum_{s \in \mathcal{S}_v} (T_{vriqjs}^B x_{vriqjs}^B + T_{vris} x_{vris}), \qquad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v.$$

   Given the results from (2) and (3), this expression in (4) is redundant and we can set

$$\begin{aligned} M_1^R = M_3^R = M_2 = & \text{ total days in the planning horizon} \\ & + \text{days of delay allowed} + \text{duration of longest voyage}. \end{aligned}$$

By following the same procedure for constraints (12.7), we obtain identical results for M_2^R and M_4^R. I.e., we also get

$$\begin{aligned} M_2^R = M_4^R = M_2 = & \text{ total days in the planning horizon} \\ & + \text{days of delay allowed} + \text{duration of longest voyage}. \end{aligned}$$

Appendix B

B.1 Confidence Levels

The numerical results presented in Chapter 14 are all based on the average of ten simulations. In this section we present the resulting confidence levels obtained from the simulations, and although ten simulations can not guarantee a certain confidence level, we see that most of the results presented in Chapter 14 has a very high statistical probability. The confidence levels was found using the Students T-Tests introduced by William Sealy Gosset in 1908. The null hypothesis for all the t-tests performed below is: The average simulated costs of the There is no difference between the two tested means, i.e. $\mu_1\mu_2$. Ten simulations performed for each strategy gives a degree of freedom of 18 for all the tests.

In Table B.1 we present the t-values and levels of confidence to reject the null hypothesis stated above and accepting the tested research hypothesis: The simulated costs obtained by the robust solutions with the NN setting are lower than the simulated costs obtained from the BASIC solution with the NN setting, i.e. $\mu_1 < \mu_2$.

Table B.1 – The t-values and confidence levels when testing that robust solutions provide costs that are lower than the BASIC strategy with the NN setting, i.e. $\mu_1 < \mu_2$. μ_1 is equal to average simulated costs obtained from the robust solutions with the NN setting. μ_2 is equal to the simulated costs obtained from the same BASIC solution with the NN setting.

μ_1	μ_2	**t-value**	**Confidence Level**
AST NN	BASIC NN	-4.0513	99.96%
REA NN	BASIC NN	-8.7126	100.00%
PRST NN	BASIC NN	-2.1116	97.55%
COMBINED NN	BASIC NN	-8.8083	100.00%

Table B.2 presents the t-values and confidence levels to reject the null hypothesis presented above when testing the following research hypothesis: The simulated costs obtained by the robust solutions with adjustable speeds are lower than the simulated costs obtained from the same robust solutions with the NN setting, i.e. $\mu_1 < \mu_2$.

Table B.2 – The t-values and confidence levels when testing that robust solutions with adjustable speed results in lower costs than with the NN setting, i.e. $\mu_1 < \mu_2$. μ_1 is equal to average simulated costs obtained from the robust solutions with adjustable speeds. μ_2 is equal to the simulated costs obtained from the same robust solution but without adjustable speeds.

μ_1	μ_2	**t-value**	**Confidence Level**
BASIC_ON	BASIC_NN	-7.7699	100.00%
AST_ON	AST_NN	-4.1244	99.97%
REA_ON	REA_NN	-1.1434	86.61%
PRST_ON	PRST_NN	-7.3172	100.00%
COMBINED_ON	COMBINED_NN	-1.2754	89.08%

In Table B.3 we present the t-values and confidence levels to reject the null hypothesis presented above when testing the research hypothesis: The simulated costs obtained by the robust solutions with adjustable speeds are lower than the simulation costs obtained from the BASIC solution, also with adjustable speeds.

Table B.3 – The t-values and confidence levels when testing that robust solutions with adjustable speeds provide lower simulated costs than the BASIC strategy with adjustable speeds, i.e. $\mu_1 < \mu_2$. μ_1 is equal to average simulated costs obtained from the robust solutions with adjustable speeds. μ_2 is equal to the simulated costs obtained from the BASIC solution with adjustable speeds

μ_1	μ_2	**t-value**	**Confidence Level**
AST_ON	BASIC_ON	-1.1795	87.32%
REA_ON	BASIC_ON	-3.6233	99.90%
PRST_ON	BASIC_ON	-3.0432	99.65%
COMBINED_ON	BASIC_ON	-4.1336	99.97%

In Table B.4 we present the t-values and confidence levels to reject the null hypothesis presented above when testing the research hypothesis: The simulated costs obtained by the robust solutions with re-planning recovery procedures enabled are lower than the simulation costs obtained from the BASIC solution with the ON setting.

Due to the small differences in the simulated costs with the ON and OR setting for the solutions of REA and COMBINED strategies, we see in Table B.4 that the level of

Table B.4 – The t-values and confidence levels when testing that robust solutions with adjustable speeds provide lower simulated costs than the BASIC strategy with adjustable speeds, i.e. $\mu_1 < \mu_2$. μ_1 is equal to average simulated costs obtained from the robust solutions with adjustable speeds. μ_2 is equal to the simulated costs obtained from the BASIC solution with adjustable speeds

μ_1	μ_2	**t-value**	**Confidence level**
BASIC_OR	BASIC_ON	-2.3456	98.47%
AST_OR	AST_ON	-1.2935	89.39%
REA_OR	REA_ON	-0.3182	62.30%
PRST_OR	PRST_ON	-1.5101	92.58%
COMBINED_OR	COMBINED_ON	-0.4205	66.04%

confidence for these two cases are only 62.30 % and 66.04 % respectively. This implies that a more thorough testing with a greater number of simulations would be required to state the research hypothesis with a greater statistical significance.

In Table B.5 we present the t-values and confidence levels to reject the null hypothesis presented above when testing the research hypothesis: The simulated costs obtained by the robust solutions with re-planning recovery procedures enabled are lower than the simulated costs obtained from the BASIC solution with the ON setting.

Table B.5 – The t-values and confidence levels when testing that robust solutions with re-planning recovery enabled provide simulated costs that are lower than the BASIC solution with adjustable speeds. μ_1 is equal to average simulated costs obtained from robust solutions with re-planning recovery enabled. μ_2 is equal to the simulated costs obtained from the BASIC solution with adjustable speeds.

μ_1	μ_2	**t-value**	**Confidence level**
AST_OR	BASIC_ON	-2.7003	99.27%
REA_OR	BASIC_ON	-4.2839	99.98%
PRST_OR	BASIC_ON	-4.8873	99.99%
COMBINED_OR	BASIC_ON	-5.2528	100.00%

The high confidence levels presented in Table B.5 imply that the null hypothesis can be rejected with a high degree of confidence with the ten simulations performed. This means that we can with confidence propose that using robustness strategies when planning and the re-planning recovery procedure during operation gives lower incurred costs than not considering uncertainty when planning and only perform speed adjustments during operation.

B.2 Development of Costs and Delays With Adjustable Speeds

Figure B.1 shows the development of incurred costs and delays during one of the performed simulations.

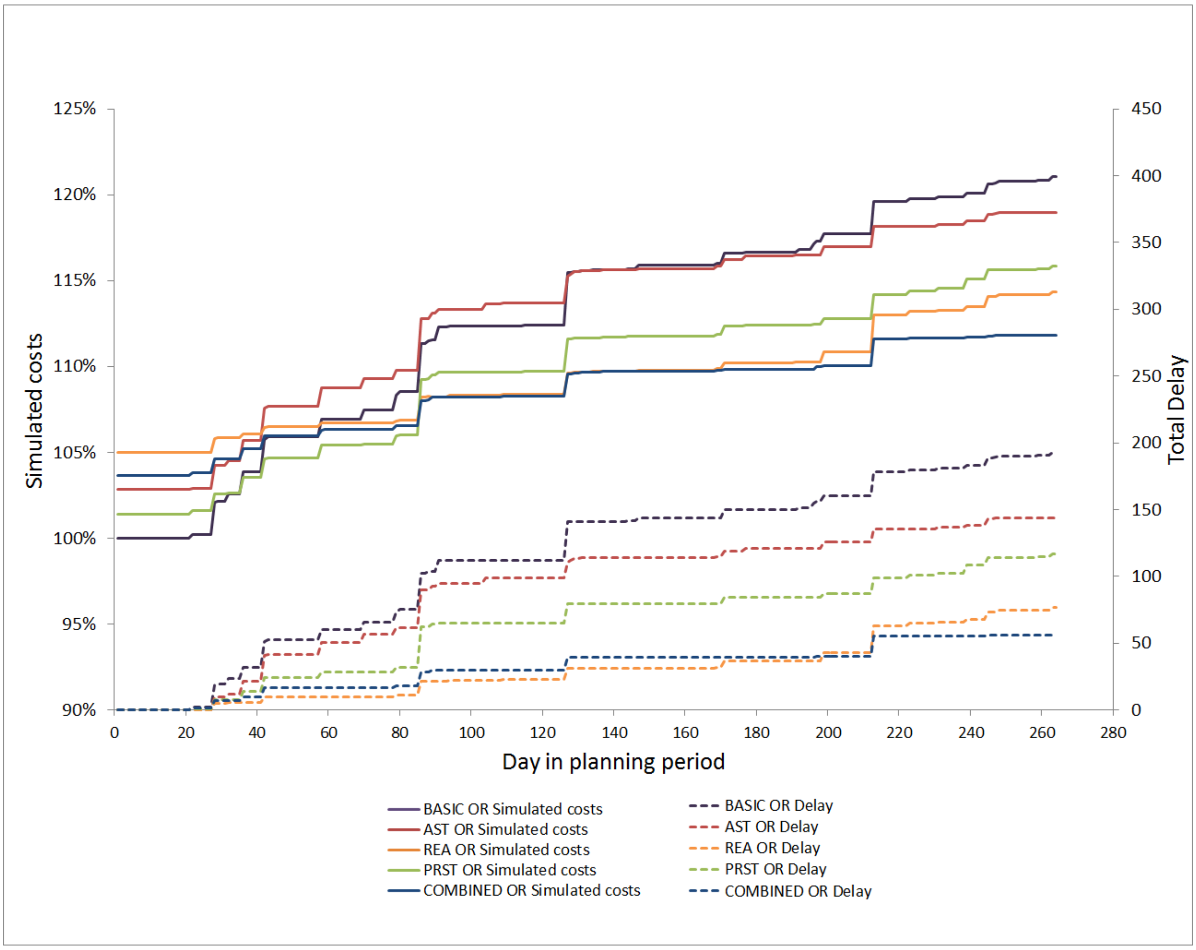

Figure B.1 – Development of simulated costs and delay in the planning period as a result of a predetermined scenario

B.3 Complete Simulations Statistics

Table B.6 shows all results obtained from the conducted NN-ON-OR simulations. The percentage values presented in Chapter 14 are calculated based on these values.

Table B.6 – Complete simulation statistics.

Method	**Planned operating costs**	**Costs caused by disruptions**	**Total costs**	**Longest delay**	**Total delay**	**Number of RP**[1]	**Voyage swaps**
BASIC_NN	2.2800E+08	1.2600E+08	3.5400E+08	17	604	0	0
BASIC_ON	2.2800E+08	4.9100E+07	2.7740E+08	9	197	0	0
BASIC_OR	2.2800E+08	3.9500E+07	2.6780E+08	7	143	12	409
AST_NN	2.3500E+08	7.0500E+07	3.0530E+08	11	329	0	0
AST_ON	2.3500E+08	3.6700E+07	2.7150E+08	7	142	0	0
AST_OR	2.3500E+08	3.1300E+07	2.6610E+08	7	116	11	300
REA_NN	2.4000E+08	2.7500E+07	2.6730E+08	9	113	0	0
REA_ON	2.4000E+08	2.2500E+07	2.6230E+08	7	82	0	0
REA_OR	2.4000E+08	2.1600E+07	2.6140E+08	7	81	9	283
PRST_NN	2.3200E+08	9.5800E+07	3.2740E+08	14	454	0	0
PRST_ON	2.3200E+08	3.1900E+07	2.6350E+08	8	112	0	0
PRST_OR	2.3200E+08	2.6600E+07	2.5820E+08	7	84	8	196
COMBINED_NN	2.3700E+08	2.8500E+07	2.6520E+08	9	118	0	0
COMBINED_ON	2.3700E+08	2.2200E+07	2.5900E+08	7	77	0	0
COMBINED_OR	2.3700E+08	2.0900E+07	2.5760E+08	7	67	7	260

[1] Re-planning recovery Procedures.

Appendix C

C.1 Attachments

The attached ZIP file contains:

1. `Fleet Deployment & Disruption Managment in RoRo liner shipping.pdf`
2. Java Files
   (a) ALNS
   (b) MIP
   (c) RHH
   (d) Simulation
   (e) SRASO
3. Output Files
   (a) Part 1
   (b) Part 2
4. Predefined Scenarios
5. Problem Instances
6. Summary of Results in Excel

C.2 How to Set Up and Run Java Files

We provide a description of how the reader may access the programmed codes, results, output files and problem instances presented in this thesis, and how they can be set up in order to duplicate the results.

To run the models or the simulation program with different robustness strategies with a specific scenario, follow the steps below:

1. In the folder Java files, choose the model or simulation to be tested.
2. Select the settings for the specified model and for the simulation, choose which robustness strategy and predefined scenario to be tested. E.g., to run the AST strategy with scenario 1 with NN setting, first open the folder called "AST" and then open the folder named "ASTNNSim1".
3. Import the Java files into the Java workspace and run the program.
4. If the MIP, RHH, ALNS or SRASO is used, the problem instance to be tested can be selected by removing and adding comment backslashes in the class `Test Instances`.
5. Make sure to have Gurobi Optimizer installed and add-ins to ensure compatibility with Microsoft Excel.
6. The output files are given in the folder called "output" in the corresponding model folder.