NTNU
Norwegian University of
Science and Technology

Modelling Approaches for Maritime Fleet Deployment with Speed Optimization and Voyage Separation Requirements

Venke Borander
Anders Straume

Industrial Economics and Technology Management
Submission date: June 2018
Supervisor: Kjetil Fagerholt, IØT
Co-supervisor: Bo Dong, IØT
Xin Wang, IØT

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

Problem Description

The purpose of this thesis is to further develop a path flow model for a fleet deployment problem within maritime transportation. The model should include voyage separation requirements, as well as speed optimization. The aim is to be able to solve larger problem instances within a reasonable amount of time. It will be implemented and tested on realistic test instances. This path flow model will then form the basis for heuristic approach, to try to solve even larger instances within reasonable time.

Preface

This master thesis has been written as part of the Master of Technology degree at the Norwegian University of Science and Technology (NTNU). The thesis is written within the field of Managerial Economics and Operations Research, at the Department of Industrial Economics and Technology Management. The work of this thesis has been conducted during the spring of 2018 and is a continuation of our Project Assignment (Borander et al., 2017) written during the fall of 2017.

The purpose of this thesis is to present and compare methods of solving the maritime fleet deployment with integrated speed optimization and voyage separation requirements.

We would like to express our deepest gratitude to our supervisor Professor Kjetil Fagerholt, co-supervisors Postdoctoral fellow Xin Wang and PhD candidate Bo Dong (all NTNU) for their academic insights and excellent guidance throughout the process of this thesis.

Trondheim, June 11, 2018

Venke Borander

Anders Straume

Abstract

This thesis studies the fleet deployment problem within the liner shipping segment of the shipping industry. A shipping company has a predefined set of intercontinental trade routes, and serves numerous voyages on each trade route within a given planning horizon. These voyages are operated by a heterogeneous fleet of ships where each ship has a predefined speed range within which it can sail. The fuel consumption, hence the fuel costs, is a function of speed. Thus, optimizing the sailing speed has a great impact of the sailing costs. The shipping companies enter Contracts of Affreightment with the cargo owners, regarding the handling of cargo. One of the things that these states is that voyages on a given trade route should be fairly even spread between them. These two factors lead to the maritime fleet deployment problem with speed optimization and voyage separation requirements.

Two models are proposed to solve the problem at hand, an arc flow and a path flow formulation. The fuel consumption function is a non-linear function of speed and is linearized by choosing discrete speed points and linear combinations of these. The voyage separation requirements are formulated as hard constraints, setting a lower limit for the required spread between voyages. The path flow model is in itself one of the main contributions from this thesis, as it has, to the authors' knowledge, never before been combined with speed optimization and the voyage separation requirement as presented in this thesis. The path flow model is based on a decomposition approach. A subproblem for each ship handles the generation of all possible paths a priori to solving the model. The master problem selects one path per ship, as well as deciding the speed along each sailing leg, in order to maximize profit. Further, path reduction heuristics are introduced. This enables the path flow model to handle larger problem instances and obtain better solutions faster.

Computational results show that the path flow model outperforms the arc flow model, both with regard to solution time and objective value. Thus, further analyses are conducted on the path flow model only. Results from analyzing the speed optimization part of the problem shows that implementing speed optimization provide higher profits, though uses longer time to find the solution. In total, the path flow model performs well, but it struggles to handle problem instances with too many paths, and the number of paths increase vastly when problem size increases. Path reduction heuristics are used to solve larger instances, that are not possible to solve within reasonable time. The path flow model with the heuristics provide a better solution quality in a much shorter amount of time. When utilizing these heuristics, problem instances with 18 ships and a planning horizon up to 150 days are solved. The effects of combining the voyage separation requirement and speed optimization are analyzed. It is discovered that including speed optimization, intensifies the need for the voyage separation requirement.

Sammendrag

Denne masteroppgaven undersøker flåteplanleggingsproblemet innen linjeshippingsegmentet av shippingindustrien. Et shippingselskap har et forhåndsdefinert sett av interkontinentale handelsruter, og i løpet av en gitt planleggingshorisont skal det gjennomføres et gitt antall seilinger på hver handelsrute. Skipssammensetningen i flåten er heterogen, og hvert skip har ett gitt hastighetsintervall det kan operere innenfor. Drivstofforbruket, og dermed drivstoffkostnadene, er en funksjon av hastighet. Dette gjør at å optimere seilingshastighetene vil ha en stor påvirkning på seilingskostnadene. Shippingselskapene inngår fraktkontrakter ("CoAs") for å utføre frakttjenester for andre selskap. En av tingene som er spesifisert i disse kontraktene er at seilinger på en gitt handelsrute skal være nokså jevnt fordelt utover i tid. Herav får vi flåteplanleggingsproblemet med hastighetsoptimering og krav om separerte seilinger.

To modeller for å løse problemet er foreslått, en arc flow og en path flow formulering. Drivstofforbruket er en ikke-lineær funksjon av hastighet og er linearisert ved å velge ut diskret hastighetspunkt og lage lineærkombinasjoner av disse. Kravet om separerte seilinger er formulert som ett absolutt krav, ved å sette en nedre grense for minimum separasjon mellom seilinger. Path flow modellen er i seg selv ett av hovedbidragene fra denne masteroppgaven, siden den, etter det forfatterne kjenner til, aldri før har blitt brukt i kombinasjonen med både hastighetsoptimering og krav om separerte seilinger. Path flow modellen er basert på en dekomponeringsmetode. Ett subproblem for hvert skip håndterer genereringen av alle mulige paths før modellen løses. Masterproblemet velger ut en path per skip, i tillegg til å bestemme hastigheten langs hver del av seilingen, med den hensikt å maksimere profitt. Videre introduseres heuristikker for å redusere mengden paths som blir generert. Dette gjør det mulig for path flow-modellen å håndtere større probleminstanser og få bedre løsninger raskere.

Resultatene fra beregningsstudien viser at path flow-modellen utkonkurrerte arc flow-modellen, både med hensyn på løsningstid og objektivverdi. Derfor er kun path flow-modellen brukt og analysert videre i studien. I analysen av hastighetsoptimering ser man at løsningstiden går opp, men at målfunksjonsverdiene forbedres når hastighetsoptimering

er implementert. Totalt sett fungerer path flow-modellen godt, men den får problemer når det blir for mange paths i probleminstansene, og antall paths i problemene øker enormt når problemstørrelsen øker. Heuristikker som reduserer antall paths i problemet, brukers for å løse større instanser som man ellers ikke ville fått løst innen rimelig tid. Når heuristikkene er anvendt for path flow-modellen oppnås bedre løsningskvalitet innen vesentlig kortere tid, enn uten. Ved å anvende slike heuristikker, har problemer på opptil 18 skip og med en planleggingshorisont på 150 dager blitt løst. Effekten av å kombinere hastighetsoptimering sammen med kravene til separerte seilinger er videre undersøkt. Dette viste at behovet for å innføre krav til separasjon av seilinger er enda viktigere når hastighetsoptimering er en del av problemet.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Maritime transport is the main distribution network for international trade and plays a key role in today's globalized world. According to the International Maritime Organization (IMO), 90% of all goods transported across borders worldwide is transported by the shipping industry. This corresponds to approximately 10 billion tons in 2015 (Asariotis et al., 2016), and the global demand has been steadily increasing over the last decades (except during the financial crisis around 2009). The shipping industry is highly connected to the macroeconomic conditions of the world. This industry is among the first to be affected in times of political and economic turmoil, by mechanisms as oil price, decrease in international trade, regulations etc. Hence, the demand may change rapidly. On the supply side, the supply of ships for the industry has a long time horizon, and is not very adaptable (Asariotis et al., 2016). Even though the growth of the world fleet has been decelerating each year for the last five years in a row, there is a tendency of overcapacity in the global fleet (Asariotis et al., 2016). In 2015, the shipping industry (with the exception of tankers) suffered from historic low levels of freight rates and weak earnings. In 2016, the container shipping segment alone reported a collective operating loss of 3.5 billion dollars (Asariotis et al., 2017). As a result, the margins are pushed down. For an industry that has high investment costs and very high daily cost rates related to operating ships, the quest for profitable operations is of higher importance than ever. One of the main targets in order to achieve this is to utilize the whole fleet capacity at all times and reduce ballast sailing and port time to a minimum. Ballast sailing, sailing without payload, is obviously very unprofitable. With this in mind, it should be possible to understand why proper planning and scheduling of maritime transportation is of high importance.

Lawrence (1972) introduced the concept of three segments of maritime transportation; industrial, tramp and liner shipping. Ronen (1983) gave a good description of each of these segments. In industrial shipping, the owners of the goods also own their own ships, and it can be compared to having your own car. Tramp shipping serves spot cargoes and follow available goods around the globe, like a taxi service. For liner shipping, the shipping companies have a set of predefined, published itineraries, and transport goods along the routes, like a bus service. For industrial shipping the goal is to minimize costs, for the other two the goal is to maximize profit. Further, planning issues in maritime transportation are categorized into strategic, tactical and operational levels (Christiansen et al., 2013). The strategic level includes decisions based on a planning horizon of years. Within shipping, this typically includes fleet composition and design, decisions regarding markets and types of cargo to transport and design of trade routes. The tactical level includes decisions with a planning horizon of months, such as ship routing and scheduling, which may be referred to as the *fleet deployment problem*. This is part of the problem that is studied in detail in this thesis. On an operational level, ship specific ship decisions are made, with a horizon of days or weeks.

Even though the fleet deployment problem has been researched and solved with optimization methods, most shipping companies solve their scheduling by manual planning. In these terms, manual planning means to schedule the fleet without the use any optimization based decision support systems. What the shipping companies do instead, is to use the scheduler's experience to set up a schedule with a simple spreadsheet (Fagerholt et al., 2010). Their planning is usually done by simplifying the problem, for example by isolating some trade routes and/or ships. This will of course have the drawback of compromising the solution quality in exchange for a problem that is easier to solve. Manual scheduling obtains feasible solutions, but are not able to utilize the fleet at the utmost or push the margins in order to maximize the profits. Today, the computational power of computers is increasing at a fierce pace, and will provide even better solutions and solve larger problems in the future. Hence, the value of implementing optimization based decision support systems will increase in the future.

This thesis concerns the fleet deployment problem within the liner shipping segment. The main task is to assign available ships to voyages on different trade routes, and aims to utilize the fleet in an optimal manner. A trade route is a predefined route from a loading region to a discharging region, while a voyage is an explicit journey on a given trade route. Trade routes are typically intercontinental, and has several port calls in the loading and discharging region of the trade route, respectively. An illustration of some intercontinental trade routes is shown in Figure 1.1. Note that a trade route does not necessarily go both ways.

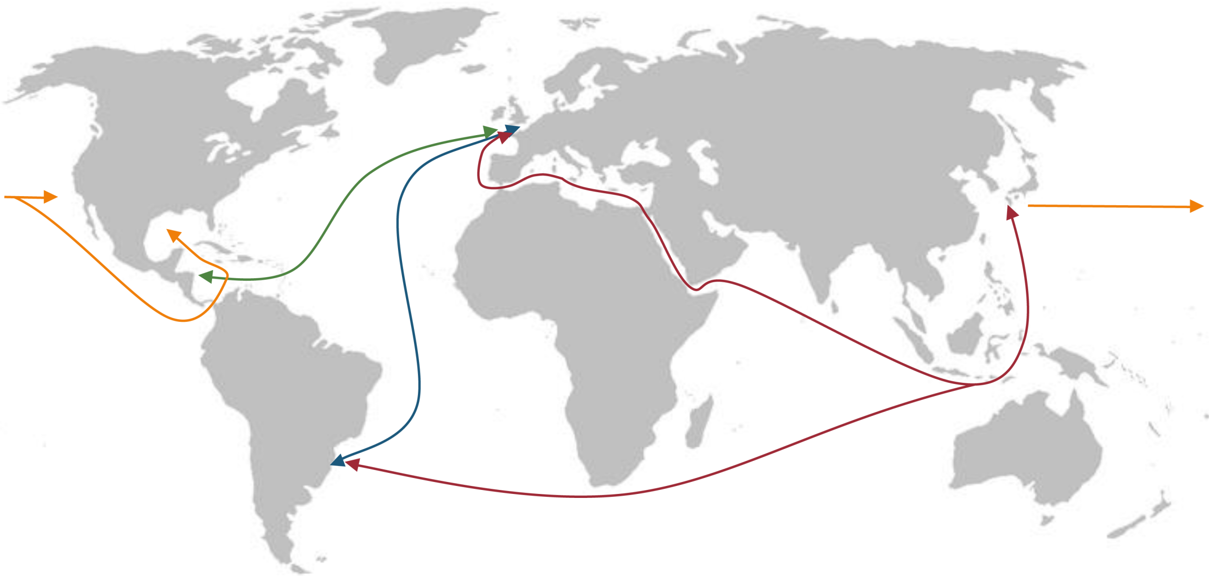

Figure 1.1: Illustration of intercontinental trade routes.

Charterers are cargo owners that enter contracts with a shipping company for the transportation of their cargo. These contracts are referred to as Contracts of Affreightment, or CoAs. The most important parts of these contracts are where the cargo is heading, the amount to be transported, at what time and the freight rate. The CoAs are long term contracts. A commonly used phrase in these CoAs, regarding the frequency of voyages on a trade route, is "fairly evenly spread". Think of a trade route that has three voyages per month. One could have predetermined a specific frequency (e.g. every tenth day) and departure dates 1st, 10th and 20th, which are perfectly spread. Instead, an alternative schedule with departure dates 3rd, 10th and 22nd, could be an acceptable fairly even schedule, and might even be more profitable. Being able to alter the dates slightly gives the shipping company some flexibility and open for a better fleet utilization, as well as lower dependency on spot ships. Spot ships are ships that are possible to charter in to serve a voyage. The use of spot ships is decided during the decision making of the fleet deployment. There may be optional voyages available as well. That is voyages the company may agree to service, in order to utilize existing overcapacity of the fleet and increase profits. The fleet deployment problem with voyage separation has been studied to some extent. Norstad et al. (2015) solved the fleet deployment problem with voyage separation for real sized problems by two approaches; an arc flow model and an a priori path generation model. Further, Vilhelmsen et al. (2017) developed a branch-and-price method as an extent to the Norstad et al. (2015) formulation. However, the problem with the addition of speed optimization is not yet investigated.

Speed has always been a crucial factor in shipping. For decades, the main target was to build faster ships to reduce sailing time and associated sailing costs. Today, the main target when reducing sailing costs is to minimize the fuel 21costs, hence the fuel consumption, which is a function of speed. Sailing speed is often assumed given within maritime optimization. The fuel consumption of a ship is also load dependent, i.e. a loaded ship on a voyage will use more fuel than when sailing ballast. For the shipping industry, fuel costs are of enormous magnitude, hence to include speed optimization opens for new and better solutions. For example, it could be profitable to speed up one voyage, in order to catch another, and thereby achieve a better, more profitable, overall solution. In practice, the captains on board already performs a sort of speed optimization. They adjust the speed to abide with the schedule and start their voyages within the time windows, and at the same time use as little fuel as possible. However, this cannot change any routing decisions, only make the best out of a given schedule. By implementing the speed optimization in the scheduling, speed optimization increases the solution space, which open up for brand new, even better overall solutions. When combining speed optimization and voyage separation, new challenges arise. Usually, when speed optimization has been implemented, the speed optimization is performed for each ship locally. Both Norstad et al. (2011) and Andersson et al. (2015), among others, handles speed optimization this way. However, the voyage separation requirements imply that the starting time of each voyage is set according to the starting times of other voyages, performed by other ships. Where the starting time of voyages and chosen speeds are highly related. Hence, there is an interdependency among the ships and their sailing speeds.

In addition to the economic effects of speed optimization, fuel consumption has a great impact on the emissions of Greenhouse Gas (GHG) as well. Transport & Environment (nd) has calculated that a reduction in speed by 10 % will reduce the total GHG emissions from the shipping industry by 19 % in total. Based on data from 2007 to 2012, the annual average emissions from the shipping industry was 1,038 billion tons CO_2 equivalents (Smith et al., 2014). This corresponds to 2,8 % of the total annual GHG emissions in the world. Cames et al. (2015) estimates that the emissions will increase to 17 % of the global GHG emissions in 2050 if no actions are taken. Smith et al. (2014) has investigated several possible scenarios regarding the future GHG emissions, where most of them are far more pessimistic. For now, there are few regulations of the emissions from the international shipping industry. However, the increased awareness of global warming and environmental changes in the global society has started to change this. Today, the emissions and environmental impact from the shipping industry are on the IMO's agenda. For example, as of January 1st 2013, new-built ships must comply with the EEDI, The Energy Efficiency Design Index (IMO, nd). The IMO's Marine Environment Protection Committee are also collecting data and working on a GHG strategy for the industry, due in 2023. In order to achieve a more environmentally friendly industry, speed optimization could be an efficient and useful tool.

This thesis investigates and compares two different approaches of solving the fleet deployment problem with voyage separation and speed optimization. Both voyage separation and speed optimization has been researched to some extent separately, whereas this thesis takes both into account. As mentioned above, incorporating both speed optimization and voyage separation at the same time causes interdependencies between ship routes, which generates a new, more complicated problem that has not been explored in any extent. The project assignment (Borander et al., 2017) looked into an arc flow formulation, and this formulation is continued in this thesis. The second approach developed in this thesis is a path flow model. Norstad et al. (2015) compared the performance of arc flow and path flow models, but without speed optimization. The addition of speed optimization is the main contribution of our thesis. Both models also take into account the different fuel consumption of laden vs. ballast sailing, and sailing speed along all sailing legs (both ballast and voyage sailing) are considered individually. The performance of the two models are compared to each other. Further, the effects of speed optimization and voyage separation are tested. This thesis also examines some heuristics rules for the path flow model, which reduces the total number of paths in the problem, hence aims to enable solving even larger problems. A number of test instances, based on data from a shipping company, are used to test the performance of the models.

The remainder of this thesis starts with a detailed problem description in Chapter 2, and Chapter 3 provides a literature review. The mathematical models are presented in Chapter 4, followed by a detailed description of the path generation for the path flow model in Chapter 5. Chapter 6 presents the instances used in the computational study in Chapter 7. In Chapter 8 concluding remarks are presented and possible further research is discussed in Chapter 9.

Chapter 2

Problem Description

This chapter gives a detailed description of the fleet deployment problem with voyage separation and speed optimization. Section 2.1 describes the general fleet deployment problem without voyage separation and speed optimization. Section 2.2 describes the speed optimization part of the problem in detail. In Section 2.3 the voyage separation requirement is described. Section 2.4 gives an example to better understand the fleet deployment problem and speed optimization. Finally, the problem is summarized in Section 2.5.

2.1 General Description

The fleet deployment problem can generally be described as the process of assigning ships to voyages on different trade routes, where the goal is to do this in a cost efficient way. The ships pick up cargo from different charterers in the origin region and unload the cargo in the destination region. The trade routes are typically intercontinental, as illustrated in Figure 2.1. Both the origin and destination region of the trade route may have several port calls. A voyage is a specific sailing along a given trade route. The trade routes in Figure 2.1, are the same ones that are used in the computational study in Chapter 7. The number of voyages on each trade route varies, and is set prior to the next planning period. The time windows for each voyage are also given.

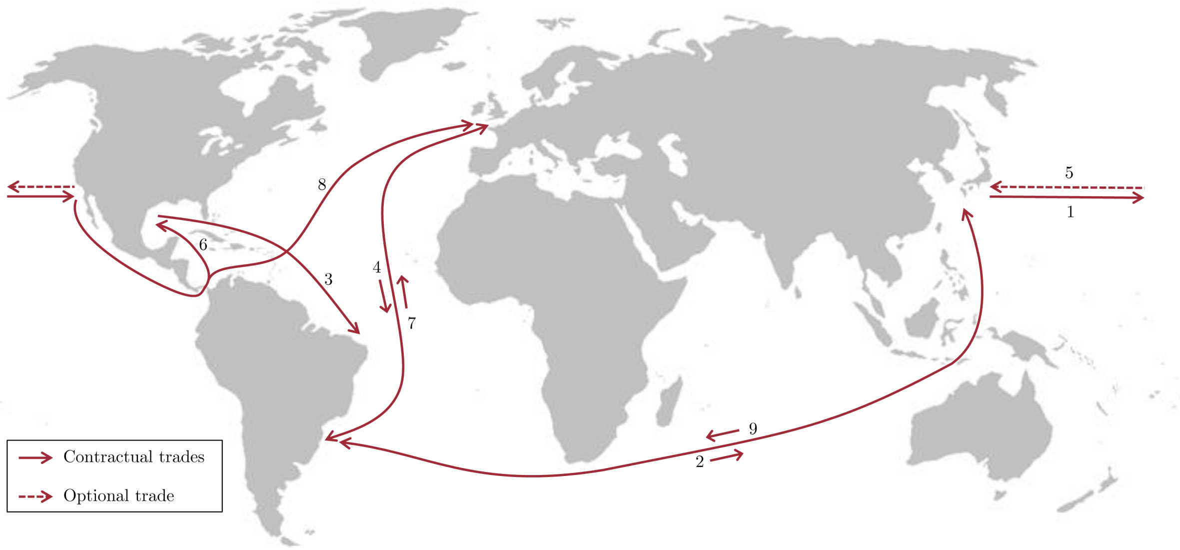

Figure 2.1: The intercontinental trade routes used in this thesis.

In general, a shipping company has a heterogeneous fleet. This means that the ships in the fleet have different properties, such as load capacities, sailing speeds, operating cost, draft restrictions etc. Due to these properties there might be restrictions on which ships that can serve the different trade routes. I.e. the ships and trade routes have to be compatible. The ships will also be accessible at different times and positions at the beginning of the planning period. Some ships are vacant at the beginning and some have to fulfill ongoing voyages, and the ships are located all around the globe. Some ships might be in dry dock or undergoing maintenance as well. Based on these arguments, all ships have to be treated individually.

The voyages are categorized into two segments, contractual and optional voyages. The shipping company is obliged to carry all contracted voyages. If the company's own fleet is not able to carry out all the contractual voyages by itself, additional spot ships can be hired to serve contractual voyages. It is assumed that the ballast sailing costs associated with chartering a spot ship is included in the charter costs. An important reason for allowing the use of spot ships is to ensure feasible solutions to the problem. Which optional voyages that are to be taken, is decided as an integrated part of the scheduling process. Each voyage has a predefined time window within which the voyage must start, instead of a fixed start-up time, which provide some flexibility for the shipping company. The contractual trade routes consist of only contractual voyages and the optional trade route consist of only optional voyages. Both optional and contractual voyages are illustrated in Figure 2.1. For longer planning horizons, the ships serve several voyages in a sequence within the planning horizon. In order to start the next voyage, a ship might have to sail ballast to re-position to the start of the next voyage.

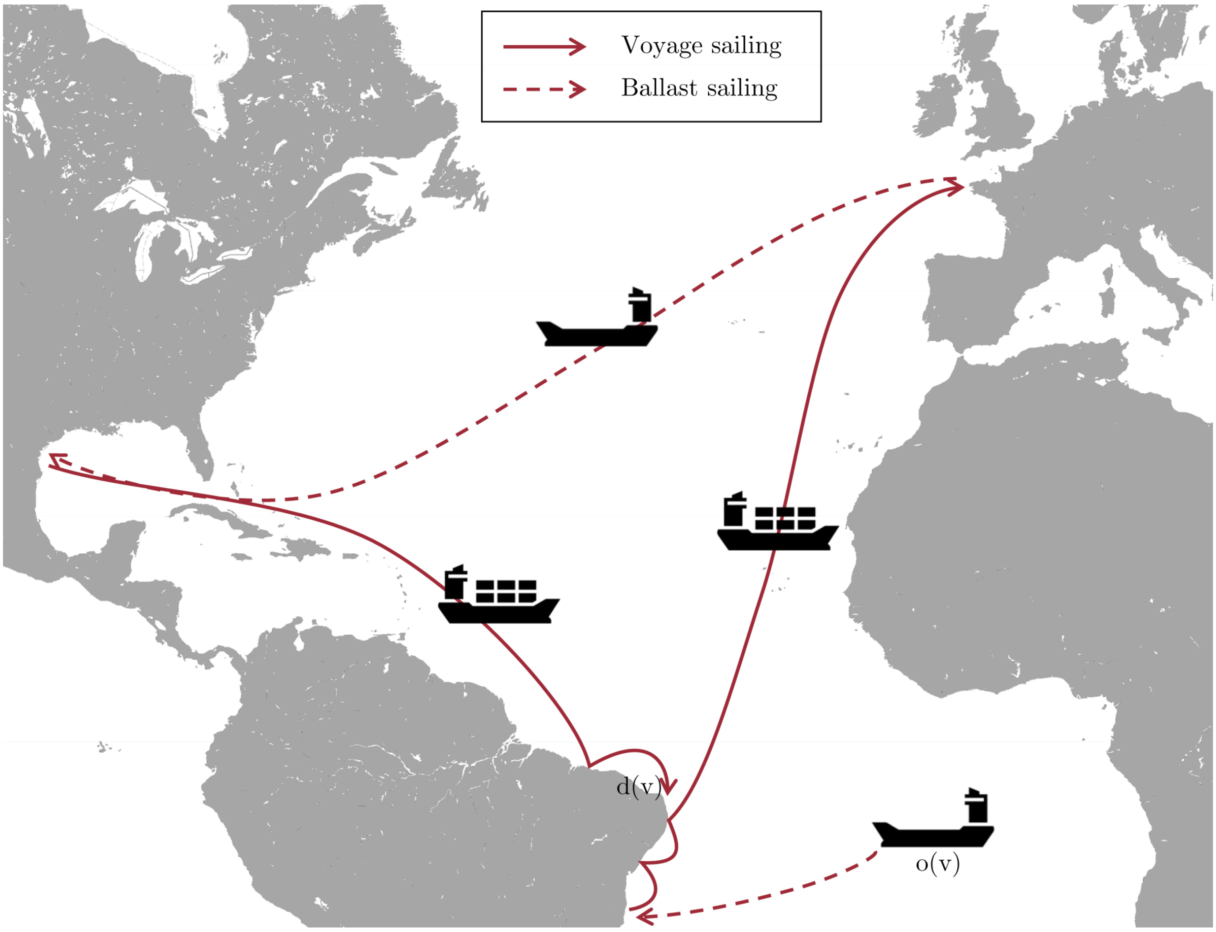

Figure 2.2: Example of sequence of sailing for a given ship.

Figure 2.2 illustrates how the sailing for a given ship may look like during a planning horizon. Here, the initial position of the ship is somewhere in the south of the Atlantic Ocean. From this location, the ship sails ballast in to South America, the origin region of the first voyage. The ship performs several port calls here, before it heads for Europe and the execution of its first voyage. The next voyage to execute has its origin in the South of the U.S., hence, the ships has to sail ballast from its current position in Europe to the south of the U.S. From here, the ship starts its second voyage, from the US and eastwards to South America, where it performs several port calls in the destination region. Please note that Figure 2.2 is an illustrative figure to showcase a possible sequence of sailing and the concepts of ballast sailing, several post calls and voyages. In reality, very long ballast sailing, as across the Atlantic Ocean, is quite undesirable and rarely used.

2.2 Fuel Consumption and Speed Optimization

The operational costs of a fleet depend heavily on fuel consumption. Therefore, a major part of optimizing the operational costs should be to optimize sailing speeds. Fuel consumption per time is approximately proportional to the cubic of the service speed. This means that the fuel consumption function is both quadratic and convex. Fuel consumption is typically a function of speed per time or distance, as shown in Figure 2.3.

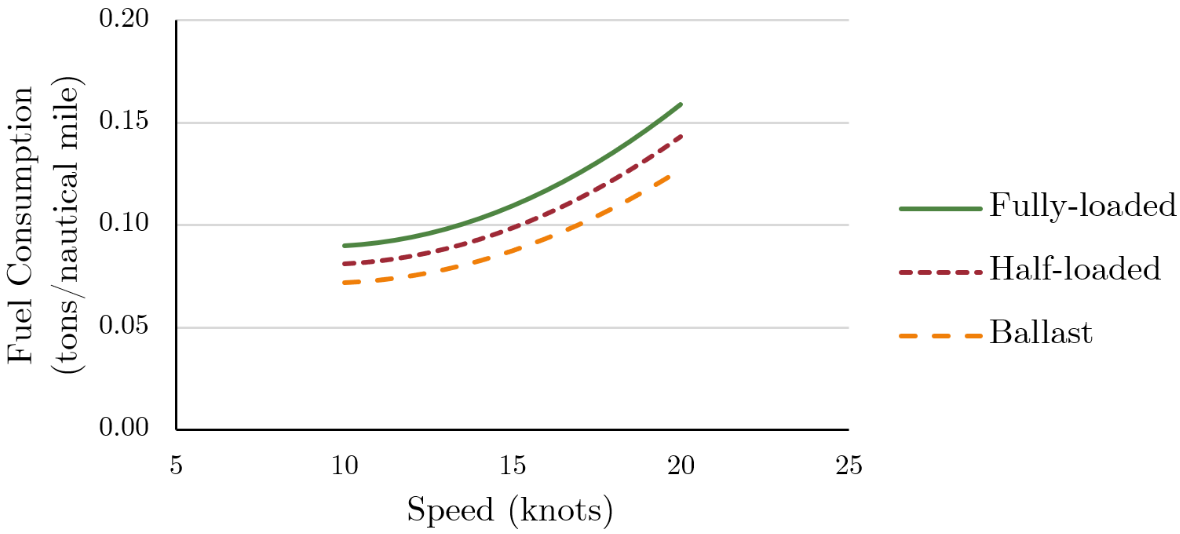

Figure 2.3: Different fuel consumption curves for different loads.

Speed optimization implies that the scheduling and fleet deployment is carried out with the possibility to adapt the operating speed for each ship on each sailing leg in the problem, instead of a fixed, preset operating speed. The aim of the speed optimization is to gain higher profits. An example where it may be beneficial to speed up is if it will reduce hiring of costly spot ships. Chartering in spot ships will incur additional costs, and it might be cheaper to increase the speed of your own ships. On the contrary, if the speed is too high, the additional cost of higher sailing speeds may be higher than the cost of chartering ships to sail those voyages. Another reason to increase sailing speed could be to take some optional voyages, given that the additional profit exceeds the added cost from increasing sailing speed. Taking optional voyages could reduce the amount of ballast sailing. If the long ballast sailing across the Atlantic Ocean in Figure 2.2 were an optional trade, instead of being quite long and costly ballast sailing, the route would be very desirable. One might also decrease the speed if a ship is well within its time window, and have no possibility of taking another voyage. With this in mind, one can understand that it is not straightforward to find the optimal sailing speed for all sailings that a ship performs during the planning horizon. The fuel consumption depends on what kind of sailing is performed, as shown by Figure 2.3, the fuel consumption is lower when sailing half-loaded or ballast. Thus, speed decisions are load dependent. Hence, all sailing legs and all ships have to be treated individually regarding speed optimization.

2.3 Voyage Separation Requirement

A common term used in the CoAs is *fairly evenly spread*, which refer to the frequency and timing of voyages on the same trade. The voyage separation requirement is highly related to this term. A voyage separation requirement state that voyages on the same trade route should be "fairly evenly spread" in time. This means that there is no absolute frequency or number of days in between two consecutive voyages on the same trade. However, the voyages on the trade should be serviced in reasonable intervals that are fairly evenly spread in time. If the term had been "evenly spread", the voyages would have to be serviced at the same intervals throughout the trade route. As this requirement is a part of the CoAs, it must be considered when deploying the fleet. Norstad et al. (2015) found that adding voyage separation requirements had minor impact on the objective value for the problem, but had great impact on the spread of the voyages, and thus, satisfying the terms in the CoAs. One way of ensuring that the *fairly evenly spread* term is satisfied is to utilize trade specific limits for separation between voyages. Such a limit would set the minimum acceptable time between two consecutive voyages on the same trade.

2.4 Example of Fleet Deployment and Speed Optimization

To better understand the problem at hand, the fleet deployment problem is illustrated by a small, numerical example. Consider a problem that consists of two trade routes, with two contractual voyages each. Each voyage has a specified time window, and the start of the voyage has to be within these time limits. The specifications for each voyage on a trade route are presented in Table 2.1. The problem is depicted in Figure 2.4 below.

Table 2.1: Specifications for the voyages in the example.

Trade	Voyage	Length (nm)	Time window (days)
1	1	7000	[5,15]
1	2	7000	[35,45]
2	1	9000	[15,30]
2	2	9000	[45,60]

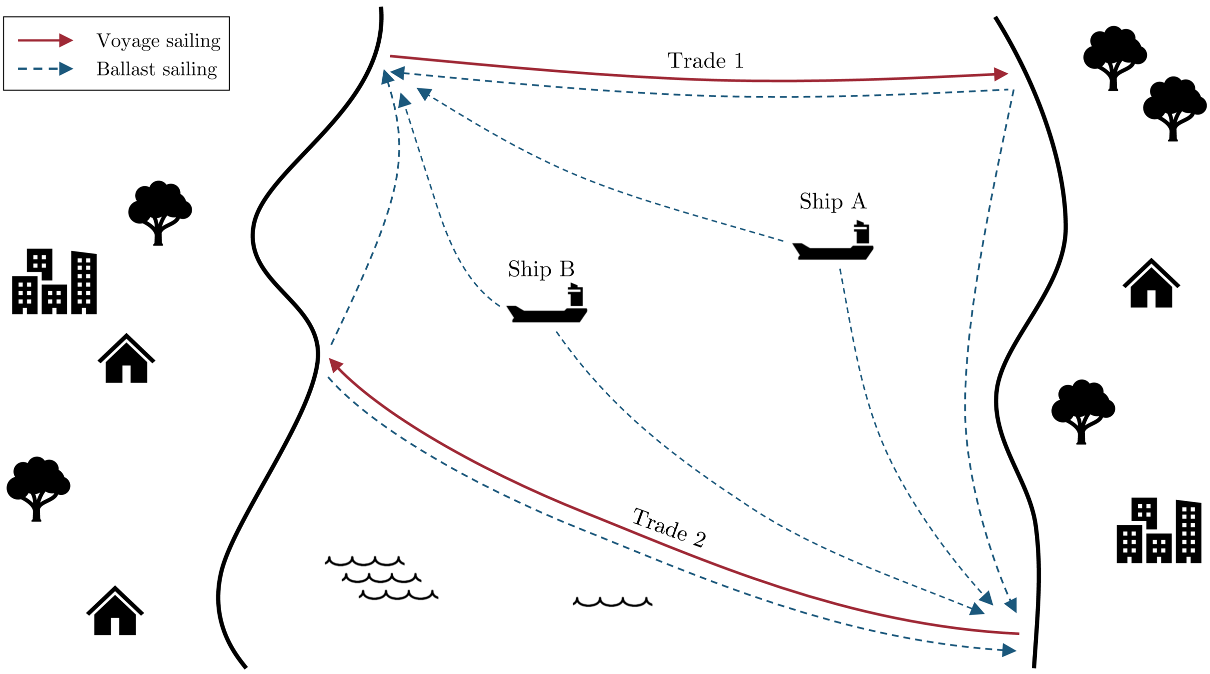

Figure 2.4: Illustration of the voyages and possible ballast legs in the example.

These four voyages are to be served by two ships from the fleet, no spot ships are available to charter in. The two ships have different origins and are available at different times, hence, they have to be treated independently. The specifications for each ship are presented in Table 2.2. The column "Available day" represents the day in the planning horizon the ship is ready to start sailing. The columns "Dist. to trade 1/2" represents the distance (in nautical miles) from a ships initial position to start region of the trade. The last column, "Speed range" represents the range of speeds that a ship can sail with.

Table 2.2: Specifications for the ships in the example.

Ship	Available day	Dist. to trade 1	Dist. to trade 2	Speed range
A	0	4000	4000	[14,20]
B	7	1500	4500	[14,20]

This problem is solved in two ways: by using the average speed as operating speed (17 knots) and by applying speed optimization, i.e. variable speed. Figure 2.5 displays all possible options for each ship from its initial position, with respect to the time windows of each voyage, *when variable speed is allowed*. When calculating the options using average speed, none of the ships are able to sail the combination $(2,1) \longrightarrow (1,2)$, and only ship A can sail $(2,1) \longrightarrow (2,2)$. Since there are no spot ships, and each ship cannot make more than two voyages, it is clear that each ship has to sail two voyages each to solve the problem.

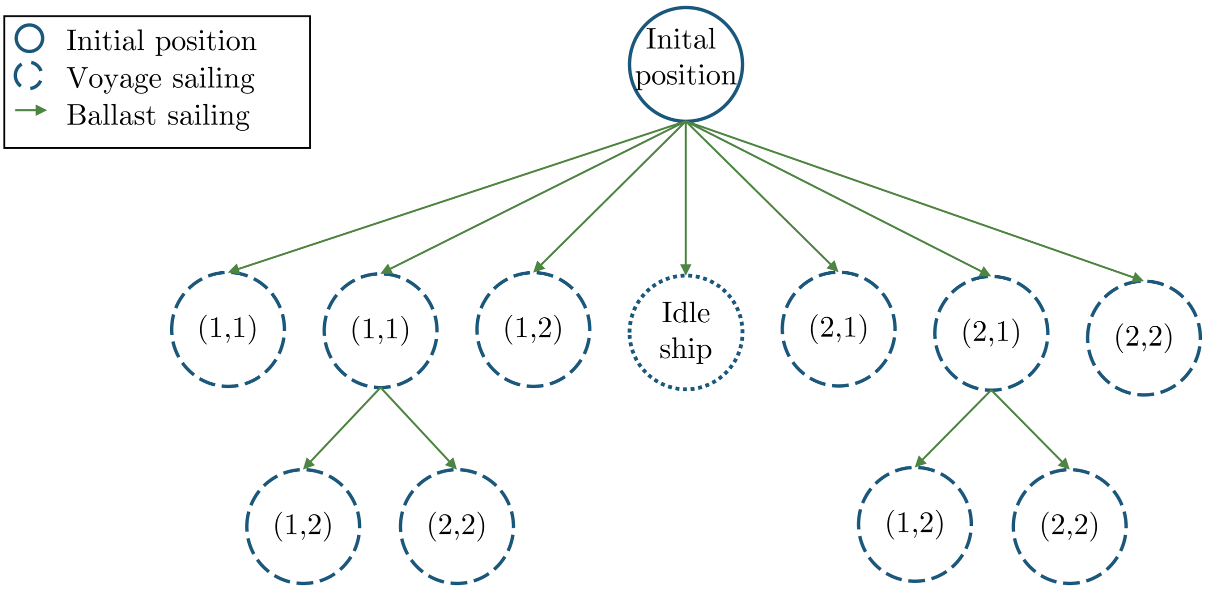

Figure 2.5: All possible sequence of sailing for each ship in the example.

When using average speed, this problem has only one feasible solution to cover all voyages: Ship A sails $(2,1) \longrightarrow (2,2)$ and ship B sails $(1,1) \longrightarrow (1,2)$. By taking the total revenue for these voyages, and subtracting the sailing costs for ballast and voyage sailing, the net profit for this solution is found to be 2,855 million USD. When applying speed optimization, this problem has four different feasible solutions, which all are within the time windows, as displayed in Table 2.3.

Table 2.3: All solutions when applying speed optimization in the example.

Solution	Ship A	Ship B	Profit
1	$(1,1) \longrightarrow (1,2)$	$(2,1) \longrightarrow (2,2)$	2,862
2	$(1,1) \longrightarrow (2,2)$	$(2,1) \longrightarrow (1,2)$	2,884
3	$(2,1) \longrightarrow (1,2)$	$(1,1) \longrightarrow (2,2)$	**2,944**
4	$(2,1) \longrightarrow (2,2)$	$(1,1) \longrightarrow (1,2)$	2,867

The profit for all these combinations are calculated in the same way as above. Solution 3 yields the best overall profits, found to be 2,944 million USD. Here, the implementation of speed optimization yields a net extra profit of 89,000 USD, a percentage increase of 3.12 %, and a new allocation of the ships of which voyages to serve.

2.5 Problem summary

The objective of the fleet deployment problem with voyage separation and speed optimization is to maximize profits, i.e. freight income minus operational costs of ships in the fleet and the chartering costs of spot ships.

There are quite a few decisions in this problem. First, one must decide what voyages a ship should sail, and in what sequence. Second, the speed on all sailing legs must be decided. Correspondingly, the start time for each voyage is decided. Lastly, one must decide which voyages (if any) are taken by spot ships.

All these decisions must comply with some constraints. All voyages must be serviced within their given time window, either by a ship in the fleet or by a spot ship. All consecutive voyages along the same trade route must be evenly spread in time.

Chapter 3

Literature Review

The literature review chapter has been divided in four parts, starting with a review of the general fleet deployment problem in Section 3.1. Section 3.2 looks into existing literature on the voyage separation requirement and other time dependencies. Literature on speed optimization in maritime transportation is reviewed in Section 3.3. The final section, Section 3.4, concerns various decomposition methods used in maritime transportation. The literature review was performed using Google Scholar and suggestions from the supervisors. Google Scholar was used by searching for keywords, for example "voyage separation" or "maritime speed optimization". These searches were assessed by reading the abstracts of the most relevant results and based on that deciding which articles to look further into. In many cases Google Scholar gave the same articles as the supervisors suggested.

3.1 Fleet Deployment in Maritime Transportation

As explained in depth in the problem description, the planning problem faced in this thesis is a fleet deployment problem. This problem has been researched to quite some extent in previous literature.

The maritime fleet deployment was concisely described by Christiansen et al. (2013) as "the tactical planning problem of assigning ships to liner routes. The planning horizon is typically a shipping season or up to 6 months." In these terms, "a shipping season" typically means a few months. However, the planning horizon can also be up to one year (Fagerholt et al., 2009).

Some of the first to propose a model to solve the fleet deployment problem were Nicholson and Pullen (1971). Their problem considers how to downscale a fleet of cargo ships. They determined the sequence in which the owned ships should be sold and to what degree the spot market for charter ships should be used. The problem was solved by using a two-stage dynamic programming model. It is divided in two stages as it would be too comprehensive to solve the entire problem with dynamic programming. Their work may be considered as the pioneering work for modelling a long-term fleet deployment problem (Gelareh and Meng, 2010).

Gelareh and Meng (2010) described the fleet deployment problem in maritime transportation as a problem which has a wide scope of applications. This means that each real-world application has its own specific features and constraints. To take care of this complication, they propose a generalized approach which most real-life applications can be derived from. This model is supposed to work as a basis for fleet deployment problems with a short planning horizon. This model has been modified by Wang et al. (2011) to eliminate combinatorial behavior in the original model.

The shipping sector that utilize models for the fleet deployment problem most, are liner and tramp shipping. These two distinct kinds of shipping cannot use the same models. Over the years there has been considerably more research on liner shipping than tramp shipping, Ronen (1983) proposed that the reason may be that the market for tramp shipping mainly consisted of small operators and the large shipping companies see the tramp market as a secondary market.

Two of the major segments within liner shipping is Ro-Ro (Roll-on Roll-off) shipping and container shipping. Hence, the most research on the fleet deployment problem has been done within these segments. In order to showcase different solution methods of the fleet deployment problem, literature regarding these segments is of interest as well. Here, literature from Ro-Ro shipping is reviewed the most. Ro-Ro shipping is the major mode for long distance intercontinental transportation of rolling equipment such as cars. Fagerholt et al. (2009) presented a new mixed integer model that avoids typical simplifications in scheduling, and allows for more flexibility than the earlier literature. They also formulated a multi-start local search heuristic to solve the problem. This heuristic was implemented in a prototype decision support system (DSS) and tested at Höegh Autoliners, a major player in the global ro-ro shipping industry. These tests showed improvements between 2% and 10% compared to manual planning. This model was extended by Andersson et al. (2015) by including speed as a variable. Andersson et al. (2015) tested their model on a case for another major company in the Ro-Ro shipping industry, Wallenius Wilhelmsen Logistics. It is fairly rare in the literature to deal with a combination of the fleet deployment problem and the inventory management problem for Ro-Ro shipping. This is exactly what Dong et al. (2017) does. This means that the problem not only handles the planning problem that a Ro-Ro company faces, but also integrates logistical services into the problem. Their results are promising, however, this is a very complex problem, and more realistic test instances could prove to be hard or impossible to solve. Fleet deployment is to some degree affected by disruptions and uncertainties. Fischer et al. (2016) propose a model with a set of robust planning strategies to handle these obstacles. Examples of such strategies are to add slack and rewarding early arrivals at ports.

3.2 Voyage Separation Requirement in Transportation

The voyage separation requirement can be modeled as either hard or soft constraints. Hard constraints set conditions for variables that are required to be satisfied. Soft constraints have some variable values that are penalized in the objective function for not being satisfied.

One way that the voyage separation requirement has been modeled is by using time windows directly. Norstad et al. (2015) used data from the Norwegian shipping company Saga Forrest Carriers to model the voyage separation requirements both as hard constraints and as soft constraints. They do this by using a parameter that determines the minimum accepted time between two consecutive voyages on a trade route. They present two models for solving a fleet deployment problem with the voyage separation requirement, an arc flow model and a path flow model with a priori path generation. The path flow model performs best when it comes to solution time, especially for large problem sizes. The a priori path generation relates to decomposition and is discussed in more detail in Section 3.4. Bakkehaug et al. (2016) use the same data as Norstad et al. (2015). However, the fleet has been expanded from 25 to 32 ships. They propose an adaptive large neighborhood search heuristic for a fleet deployment problem with voyage separation requirements. The voyage separation requirement is modeled as the minimum time elapsed between two consecutive sailings on a trade, and yields solutions with voyages fairly evenly spread over the planning horizon. This model uses a heuristic approach. Vilhelmsen et al. (2017) base their article on the same data as Bakkehaug et al. (2016) and Norstad et al. (2015). However, their method is a Branch-and-Price procedure and uses a dynamic programming algorithm to generate columns. This method is an exact method. The voyage separation requirements are relaxed in the master problem. They use a time window branching scheme to enforce these restrictions. Comparing their results with Norstad et al. (2015) they state that their model is significantly faster than Norstad et al. (2015)'s a priori path generation method, except for one instance.

Another way that modelling voyage separation has been done, is by using predefined patterns. Sigurd et al. (2005) include time separation requirements on recurring visits to the same port in their general pickup and delivery problem. They are doing this by generating predefined patterns which include time separation requirements. The mathematical model is restricted to only choose one of these patterns for each customer. Halvorsen-Weare et al. (2012) include spread of departures by the same principle as Sigurd et al. (2005) in their ship planning problem for supply ships in the offshore segment. In their model they generate all possible patterns of voyages including departure times from the supply depot before solving the model. Both these examples with evenly spread restrictions require that the number of ports to visit during the planning horizon is known in advance.

Within other modes of transportation and scheduling there can be found several examples of a voyage separation requirement, or some other different time dependency. In air transportation the voyage separation requirement has been enforced by using time windows directly by Bélanger et al. (2006) and penalizing short spacing between consecutive flights that serve the same origin-destination pair of airports. Anticipated profits depend on the schedule and the selection of aircraft types. In other modes of transportation and scheduling, there may be a need to synchronize events rather than to separate them. Even though synchronization and separation are opposites, the basic optimization idea for achieving synchronization and separation are similar, as they both connect the timing of events in some manner. Separation methods may require minimal changes to apply to separation and vice versa. Thus, articles on synchronization should also be considered when researching voyage separation. Optimization of schedules in the health sector is a problem that often has to consider synchronization requirements. Redjem et al. (2012) consider the problem of coordinating health visits at patients' homes. They solve this, in the same way as the separation problems described earlier, by utilization of time window constraints. They solve the problem as a bi-criteria problem where the time is considered in one of the objective functions. Borsani et al. (2006) also consider the problem of synchronizing resources for health care to patients at home. They solve it as a multi-objective model utilizing penalties if for example outsourcing is necessary or visits is necessary a time when it is not preferred. In the vehicle routing problem with time windows (VRPTW), synchronization and precedence constraints has been considered by Dohn et al. (2011). They presented two Dantzig–Wolfe reformulations of two compact formulations and proposed four master problem formulations.

3.3 Speed Optimization in Maritime Transportation

Most of the models found in the maritime transportation literature assume fixed and known speeds for the ships, either as implicit or explicit input (Psaraftis and Kontovas, 2014). As mentioned in the Section 2.2, the fuel consumption is a non-linear relation of speed. Many papers, among others Ronen (1982), assume that daily fuel consumption is a cubic function of ship speed (Non-linear, quadratic convex, cubic convex). This section intends to review the different takes on the relationship between fuel consumption and speed.

Andersson et al. (2015) use a linear combination of predefined discrete speed alternatives and interpolation in order to provide the desired fuel consumption as a piece-wise linear function of speed. They propose a new linearized modeling approach for integrating a speed optimization when planning ship routes. They also set different speeds for ballast and laden sailing, as the fuel consumption is different. Fagerholt et al. (2015) use the linearization method described by Andersson et al. (2015) to determine which speed to set on routes between predefined ports. Their main focus regard the regulations set by the Emission Control Areas (ECAs). To comply with these regulations, they set a different fuel price whether a route is inside or outside these areas. The model used by Andersson et al. (2015) is an arc flow model, like the model used by Norstad et al. (2015), but includes speed optimization and does not consider voyage separation. As mentioned, Norstad et al. (2015) also presented a path flow model. Wen et al. (2016) also use a path flow model to solve the fleet deployment problem, but in addition they also integrate speed optimization in the problem. They consider a different fuel consumption for different loads and set different speed for the legs of a route. They linearize the speed in a similar fashion to Andersson et al. (2015). Although they integrate speed in the model, they consider it to contribute a severe negative development to the solution time, and therefore only solve the model using a heuristic.

Another approach to speed optimization can be found in Wang and Meng (2012). They use an outer-bound approximation to linearize the fuel consumption function, adds linear constraints to bound the fuel consumption and thereby getting a linear optimization problem that is solved by a commercial solver (CPLEX). This method underestimates the fuel consumption. However, the linearization algorithm used by Wang and Meng (2012), ensures a linearization at least as good as the actual consumption within a given margin of error. With this linearization, they then proceed to investigate the optimal sailing speed of container ships on each leg of each ship route in a liner shipping network, while considering transshipment and container routing. Reinhardt et al. (2016) also solve the speed optimization problem for container shipping with transshipment, but consider the transit time of containers in addition. They solve speed optimization of an existing liner shipping network by adjusting the port berthing times. The changes of port berth times are only accepted if they lead to savings above a threshold value, which is set by a penalty parameter. The fuel consumption function is linearized by using tangent lines along the function. In contradiction to Wang and Meng (2012) the number of tangents is an input to the model. This model is essentially rescheduling existing port reservations. The model is called The Liner Shipping Berth Scheduling with Transit Times Problem (BTRSP). They model the fuel consumption as a cubic function. The advantage of this model is that it can be applied to large instances and solved within a fairly small amount of time.

The fuel consumption function can also be linearized more directly through discretization. Fagerholt et al. (2010) solve the speed optimization problem by discretizing arrival times and solving it as a shortest path problem on a directed acyclic graph. The speeds that are optimized are from an existing network of trades. Norstad et al. (2011) solve the problem of speed optimization in tramp shipping, and continue the work of Fagerholt et al. (2010) by using the same method of approximating fuel consumption. In addition, they introduce the speed on each sailing leg as a variable and using a multi-start local search heuristic to solve the problem. This gives a non-linear factor in the objective function as well as in time-related restrictions. They also present another algorithm to linearize the speed, utilizing the fact that it makes sense for a ship to have the same speed on consecutive voyages. In order to ensure that this is possible they disregard the time windows. They call this algorithm a recursive smoothing algorithm (RSA) and has zero gap. However, this is achieved by assuming that each ship has the same fuel consumption for both laden and ballast sailing. Hvattum et al. (2013) present proofs of this algorithm. They prove that the algorithm provides optimal solutions, that it gives exact solutions and that it has a worst case running time of $O(n^2)$. The speed can also be discretized by splitting the fuel consumption function in small intervals where the cost is nearly unchanged. This method is what Gelareh and Meng (2010) used when they linearized the speed. This means that every time interval is assigned one cost corresponding to the start of the interval. They use this linearization as part of a fleet deployment problem. Alvarez et al. (2010) consider, like Reinhardt et al. (2016), the problem of berthing in ports, but they look at it from the viewpoint of the ports. This means that their problem is to optimize which ship is assigned at a berth, ensuring that all land-side equipment(LSE) is available and optimizing speed such that a berth is available when the ship arrive. For solving the problem, they utilize an approach which is a hybrid of simulation and optimization. The model does not linearize the fuel cost function, but does rather use discretization through simulation.

All articles presented in this section consider speed optimization, but only a small portion of them integrate the speed optimization in the fleet deployment model.

3.4 Decomposition Methods in Maritime Transportation

A common solution procedure when dealing with a large problem is to decompose the problem into a master problem and subproblems. This is in most cases easier than solving the original problem. This idea is what resulted in the column generation method. A good example of when one would use column generation is a scheduling problem where one has to schedule events for a multiple of participants. Examples of this kind of problem is the nurse scheduling problem and the multi-vehicle routing problem. These are problems where the main problem can be easily decomposed in to subproblems for each participant that are fairly easy to solve. The subproblem finds the most promising solutions. Thereafter the master problem finds the optimal solution among the schedules found by the subproblem. The subproblems can be solved heuristically, while the master problem is solved exactly.

Decomposition approaches has been used in maritime routing and scheduling. Christiansen et al. (2004) stated in their review of ship scheduling routes for industrial shipping that as much as 40 per cent of the reviewed problems had been solved by utilizing set partitioning. The major reason for decomposition approaches being successful in maritime transportation is that the problems are often tightly constrained and thus it is possible to generate all cargo combinations for all ships a priori (Christiansen et al., 2007). Maritime transportation problems that are solved with a decomposition approach are typically solved in a similar fashion to what Fisher and Rosenwein (1989) did. They faced the problem of efficiently scheduling pickup and delivery of bulk cargo. They solved the problem by generating a menu of all feasible schedules to find the optimal solution. They also introduce limitations to the generation through heuristically limiting the menu to the most likely schedules. As mentioned previously, Norstad et al. (2015) and Vilhelmsen et al. (2017) also utilized a decomposition approach to solve the fleet deployment problem. Norstad et al. (2011) discovered, as expected, that the method that utilized a decomposition approach was superior to when the problem was solved straightforward.

Even though there are many examples of decomposition approaches being used to solve the fleet deployment problem, there are very few articles which also include speed optimization. Wen et al. (2016) is the only article that the authors were able to find that utilize a combination of set partitioning and speed optimization to solve the fleet deployment problem. As mentioned earlier they do not solve the problem exact, as they consider the inclusion of speed optimization to be a severe negative contribution to the solution time as the problem size is already quite large. They instead develop a heuristic branch-and-price algorithm to solve the problem.

Chapter 4

Mathematical Models

In this section, the mathematical formulations for the fleet deployment problem with speed optimization are given. First, a presentation of the linearization of the fuel consumption function is given in Section 4.1, before the arc flow and path flow models are presented in Sections 4.2 and 4.3, respectively. Some aspects regarding modelling choices are presented in section 4.4.

4.1 Linearization of Fuel Consumption

In order to perform speed optimization, it is necessary to look into the non-linear relationship between sailing speed and fuel consumption. The fact that the fuel consumption function for a ship is a non-linear function of speed leads to non-linearities in the objective function, in the terms related to sailing cost. The fuel consumption function is typically estimated as a cubic function of speed in tons per time. When considering the fuel consumption as a function of speed in tons per distance, it is a quadratic function. As mentioned in Section 3.3, one approach to approximate these non-linearities is to linearize the fuel consumption function. This section presents one such approach for linearizing the fuel consumption. In the project assignment, two additional approaches for linearizing the fuel consumption function were tested. The linearization approach presented in this section is the same as the one used by, among others, Andersson et al. (2015) and

Fagerholt et al. (2015). The way the non-linearities is handled by this approach, is by using discrete speed alternatives and interpolations between these. This is done by using selected speed points from the nonlinear function and allocating weights to these speed points. When analyzing different linearization approaches in the project assignment, this linearization worked best.

The linearization is performed with two and three discrete speed points, respectively. The exact number of speed points will be chosen and explained in further detail in the computational study. Figure 4.1 illustrates a fuel consumption curve, along with the two linearizations. In order to see the difference between the three curves, the illustration is exaggerated. Within the relevant speed range, the fuel consumption function is a convex curve. Therefore, the piece-wise linearization causes an overestimation of the fuel consumption compared to the original curve. This is caused by the fact that any linear combination of two points will always be strictly above the convex function between these two points, as shown in Figure 4.1.

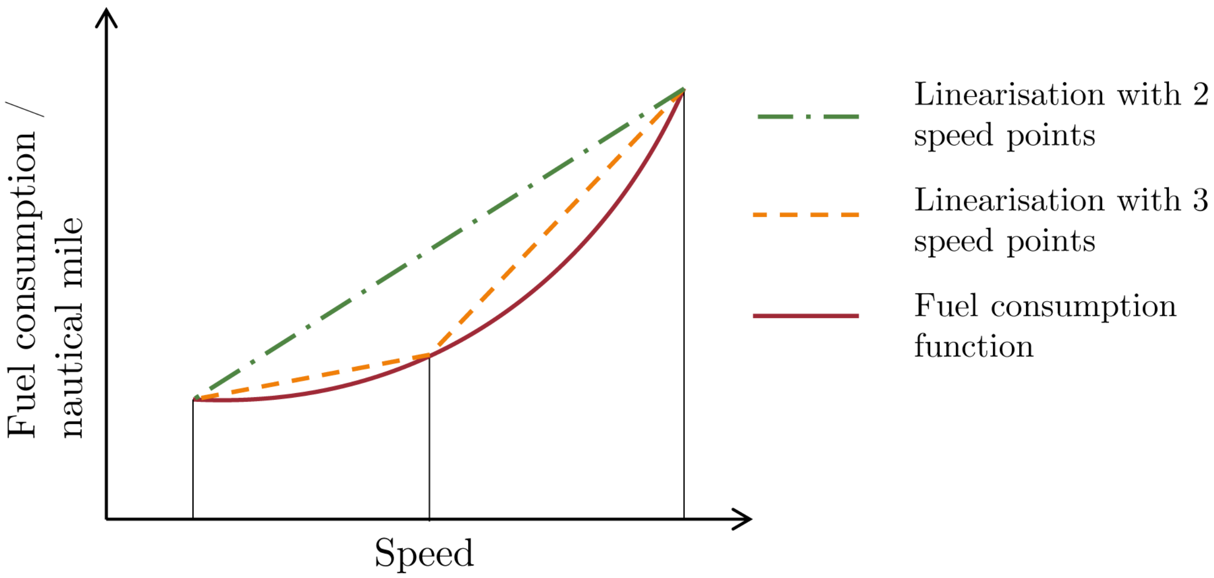

Figure 4.1: Linearizations of the fuel consumption function.

Solving the model with only one speed point is the same as solving the model without speed optimization, and the speed is fixed when the routing decisions are taken. In regular fleet deployment (without speed optimization) the operating speed is usually set as the average speed. Norstad et al. (2011) showed that using the maximum speed as operating speed during the scheduling gave better solutions than using the average. Another advantage of using the maximum speed with only one speed point, is to get the same solutions space as for two and three speed points, respectively. However, it is crucial to apply a posteriori speed optimization, to obtain a fair comparison between the models. To do a posteriori speed optimization means that the problem has to be re-optimized with respect to speed. This is done by taking the fleet deployment given by the optimal solution, and then optimize the sailing speeds along all sailing legs, without changing any routing decisions and still abide the time windows. A posteriori speed optimization is performed with a high number of discrete speed points for the fuel consumption linearization. This ensures a very close linearization of the fuel consumption function compared to the real curve, which gives negligible deviations. Hence, the profit obtained by the a posteriori speed optimization gives a virtually real profit.

4.2 Arc Flow Model

This model is an altered version from the project assignment. The model is based on the arc flow model formulated by Andersson et al. (2015). The speed optimization in our model is somewhat more advanced. This model is similar to the ones presented by Norstad et al. (2015) and Vilhelmsen et al. (2017), but the addition of speed optimization makes it more complex. This model gives a full description of the optimization problem. A compact summary of the notation and model may be found in appendix A.

4.2.1 Notation

Let $\mathcal{V}$ be the set of all ships in the fleet of the shipping company, indexed by v. The ships have individual starting positions and maintenance schedules, and are therefore treated individually, as treating them as a group could lead to infeasible solutions.

The set $\mathcal{R}$ denotes the set of all trade routes operated by the company, indexed by r. $\mathcal{R}_v$ is a subset of $\mathcal{R}$ for which trade routes ship v can carry out. Let the set $\mathcal{I}_r$={1,2,3, ... n_r} be the set of voyages on trade route r, where n_r is the number of voyages on trade route r

that has to be performed during the planning period. The set of voyages is indexed by i. The given problem can be formulated on a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ contains all nodes, and $\mathcal{A}$ is the set of arcs. $\mathcal{N}$ consists of four different kinds of nodes: Origin nodes, destination nodes, voyage nodes and maintenance nodes.

Each voyage is identified by its trade route, r, and the voyage number, i, on that trade route, (r,i). For each ship v in the set $\mathcal{V}$, the origin node $o(v)$ in set $\mathcal{N}$ represents the initial starting position that the ship is available from. The destination node $d(v)$ in set $\mathcal{N}$ corresponds to an artificial ending position. The artificial destination does not exist physically, but is in the same position as the final port of the last voyage sailed, i.e. the distance from any node to the destination node is zero. The set $\mathcal{N}^C$ is a subset of $\mathcal{N}$ and represents all the contracted voyages that the company is required to service. The set $\mathcal{N}^O$ is also a subset of $\mathcal{N}$ and represents all the optional voyages. The set $\mathcal{N}_{v}$,, a subset of $\mathcal{N}$, consists of all nodes that ship v is compatible with, in other words, the nodes that ship v can service. The set $\mathcal{N}_v^M$ is the set of required maintenance nodes for ship v. For all, but the ships that are due for maintenance, the set $\mathcal{N}_v^M$ will be empty. If ship v is due for maintenance, the ship has to visit one, and only one, maintenance node during the planning period. It is also necessary to define a set of speed alternatives, this set is called $\mathcal{S}$. The speeds in this set are ordered from lowest to highest speed.

The set $\mathcal{A}$ represents all arcs. The arc $((r,i),(q,j))$ corresponds to sailing ballast directly from the end of voyage node (r,i) to the start of voyage node (q,j). The arcs from the origin nodes to voyage nodes, $((o(v)),(r,i))$, and the arcs from voyage nodes to the destination nodes, $((r,i),(d(v)))$, are also included in $\mathcal{A}$. If maintenance is required, the maintenance nodes are treated in the same way as the voyage nodes, as described above. The set $\mathcal{A}_v$ consists of all arcs that ship v can service, that is, the set of arcs such that the ship v can sail directly from node (r,i) to node (q,j). If a ship services the arc $(o(v),d(v))$ it sails directly from the starting node to the ending node, thus the ship is idle and is not used at all.

Let T^B_{vriqjs} be the time it takes to sail ballast from the last discharge port of voyage (r,i) to the first loading port of voyage (q,j), in other words sailing the arc $((r,i),(q,j))$, with speed alternative s. The corresponding cost to this parameter is C^B_{vriqjs}. The time it takes to sail ballast from the starting position to start of voyage (r,i) with speed alternative s is $T^B_{vo(v)ris}$, and the corresponding cost is $C_{vo(v)ris}$. The time it takes to sail voyage (r,i) with speed alternative s is denoted by T_{vris}, which corresponds to sail a voyage, including the service time of all ports. The corresponding cost is C_{vris}. The estimated freight income minus the port costs, for sailing voyage (r,i) is R_{ri}. C^S_{ri} is the cost of chartering a ship from the spot market to service voyage (r,i). Each voyage has to start at its first port within a given time window, $[E_{ri}, L_{ri}]$. The parameter E_{ri} is the earliest time for starting voyage i on trade r, while L_{ri} is the latest time for starting the voyage. Let $E_{o(v)}$ be the earliest time ship v can start from its initial position. Let B_r be the minimum acceptable time between two consecutive voyages on trade r.

Let x_{vriqj} be a variable, which is 1 if ship v travels directly from node (r,i) to node (q,j), otherwise it is 0. The variable $x_{vo(v)ri}$ is 1 if ship v travels from it starting position to node (r,i), otherwise it is 0. Let $x_{rid(v)}$ equal 1 for ship v if (r,i) is the last node it services, and 0 otherwise. Similarly, $x_{o(v)d(v)}$ is 1 if ship v is idle, and 0 otherwise. Let u^S_{ri} be 1 if voyage i on trade r is serviced by a chartered ship, and 0 otherwise. All x- and u-variables are binary. The time for start of voyage i on trade r is defined by the variable t_{ri}. Variables for determining the weight of speed alternatives for a voyage and the ballast sailing to the next voyage is also necessary. Let w^B_{vriqjs} be the weight of speed alternative s for sailing ballast to the starting port of the voyage (q,j) from the end of voyage (r,i) for ship v. Let w_{vris} be the weight of speed alternative s for sailing voyage (r,i) for ship v. Let $w^B_{vo(v)ris}$ be the weight of speed alternative s for sailing ballast to the starting port of the voyage (q,j) from the origin node $o(v)$ for ship v.

4.2.2 The Model

Objective function

$$\max \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{s\in\mathcal{S}}(R_{ri}-C_{vris})w_{vris}$$
$$-\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{s\in\mathcal{S}}C^B_{vo(v)ris}w_{vo(v)ris} \tag{4.1}$$
$$-\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}\sum_{s\in\mathcal{S}}C^B_{vriqjs}w^B_{vriqjs}+\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{I}_r}(R_{ri}-C^S_{ri})u^S_{ri}$$

The objective function (4.1) maximizes profit by summing the combination of the most profitable voyages for the fleet and the most favorable spot ship options.

Service constraints

$$\sum_{v\in\mathcal{V}_r}\left[\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}x_{vriqj}+x_{vrid(v)}\right]+u^S_{ri}=1,\quad (r,i)\in\mathcal{N}^C \tag{4.2}$$

$$\sum_{v\in\mathcal{V}_r}\left[\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}x_{vriqj}+x_{vrid(v)}\right]\leq 1,\quad (r,i)\in\mathcal{N}^O \tag{4.3}$$

$$\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}x_{vriqj}+x_{vrid(v)}=1,\quad v\in\mathcal{V},(r,i)\in\mathcal{N}^M_v \tag{4.4}$$

Constraints (4.2) state that each contracted voyage must be serviced by either a ship in the fleet or a spot ship. Constraints (4.3) state that each optional voyage can be serviced at most once by a ship within the fleet. Constraints (4.4) ensure that all maintenance operations are performed.

Network flow constraints

$$x_{vo(v)d(v)}+\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}x_{vo(v)ri}=1,\quad v\in\mathcal{V} \tag{4.5}$$

$$x_{vrid(v)} + \sum_{q \in \mathcal{R}_v} \sum_{j \in \mathcal{I}_q} x_{vriqj} - \sum_{q \in \mathcal{R}_v} \sum_{j \in I_q} x_{vqjri} - x_{o(v)ri} = 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.6}$$

$$x_{vo(v)d(v)} + \sum_{r \in \mathcal{R}_v} \sum_{i \in \mathcal{I}_r} x_{rid(v)} = 1, \quad v \in \mathcal{V} \tag{4.7}$$

$$x_{vo(v)ri} - \sum_{s \in \mathcal{S}} w_{o(v)ris} = 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.8}$$

$$x_{vriqj} - \sum_{s \in \mathcal{S}} w_{vriqjs} = 0, \quad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \tag{4.9}$$

$$x_{vrid(v)} + \sum_{q \in \mathcal{R}_v} \sum_{j \in \mathcal{I}_q} x_{vriqj} - \sum_{s \in \mathcal{S}} w_{vris} = 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.10}$$

Constraints (4.5)-(4.7) ensure network flow for each ship. Constraints (4.5) state that a ship must either be idle or leave its starting position to a node (r,i), while constraints (4.7) state that a ship must either be idle or arrive at its ending position from a node (r,i). Constraints (4.6) ensure that each voyage starts in an origin node, that every node entered into is also exited, and that each voyage ends up in a destination node. Constraints (4.8)-(4.10) describe the relation between the flow variables and the variables for weighting speed alternatives for initial sailing, ballast sailing and voyage sailing respectively. The weights of the speed alternatives should sum up to 1 if a voyage is serviced by that ship. If the voyage is not serviced by that ship, the weights should add up to 0.

Time constraints

$$x_{vo(v)ri}(E_{o(v)} + \sum_{s \in \mathcal{S}} T^B_{vo(v)ris} w_{vo(v)ris} - t_{ri}) \leq 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.11}$$

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

$$E_{ri} \leq t_{ri} \leq L_{ri}, \quad r \in \mathcal{R}, i \in \mathcal{I}_r \tag{4.13}$$

Constraints (4.11) ensure that time spent sailing from the initial position to the first voyage (r,i) does not exceed the last start time for voyage i. Constraints (4.12) state that the time spent on voyage (r,i) and ballast sailing to the start of voyage (q,j) does not exceed the latest starting time of voyage (r,i). Note that, constraints (4.11) and (4.12) are non-linear and has to be linearized in order to solve the problem. Constraints (4.13) secure that the time window for each voyage is not violated.

Evenly spread constraints

$$t_{r,i+1} - t_{ri} \geq B_r, \quad r \in \mathcal{R}, i \in \mathcal{I}_r \backslash \{n_r\} \tag{4.14}$$

Constraints (4.14) are hard, evenly spread constraints of consecutive voyages on the same trade. These constraints ensure a minimum time spread between consecutive voyages in accordance to the voyage separation requirement.

Binary and Non-negativity Constraints

$$x_{vo(v)d(v)} \in \{0,1\}, \quad v \in \mathcal{V} \tag{4.15}$$

$$x_{vo(v)ri} \in \{0,1\}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.16}$$

$$x_{vrid(v)} \in \{0,1\}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.17}$$

$$x_{vriqj} \in \{0,1\}, \quad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \tag{4.18}$$

$$w_{vo(v)ris} \in [0,1], \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, s \in \mathcal{S} \tag{4.19}$$

$$w_{vris} \in [0,1], \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, s \in \mathcal{S} \tag{4.20}$$

$$w_{vriqjs} \in [0,1], \quad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v, s \in \mathcal{S} \tag{4.21}$$

$$t_{ri} > 0, \quad r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.22}$$

$$u^S_{ri}, \in \{0,1\} \quad r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.23}$$

Constraints (4.15)-(4.23) define all variables as either binary, weighing or continuous variables.

4.2.3 Linearization of Non-linear constraints

As mentioned above, the time constraints (4.11) and (4.12) need to be linearized. This is done by the well know big-M method. The big-M parameters are calculated as follows:

$$M_{vri} = E_{o(v)}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.24}$$

$$M_{vriqj} = L_{ri}, \quad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \tag{4.25}$$

Constraints (4.11) and (4.12) are linearized using the big-M parameters. As the big-M parameters only consist of one parameter, these parameters are used directly to achieve the following new constraints:

$$\begin{aligned} &E_{o(v)} + \sum_{s \in \mathcal{S}} T^B_{vo(v)ris} w_{vo(v)ris} - t_{ri} - E_{o(v)}(1 - x_{vo(v)ri}) \leq 0, \\ &v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \end{aligned} \tag{4.26}$$

$$\begin{aligned} &t_{ri} + \sum_{s \in \mathcal{S}} (T_{vris} w_{vris} + T^B_{vriqjs}) w_{vriqjs} - t_{qj} - L_{ri}(1 - x_{vriqj}) \leq 0, \\ &v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \end{aligned} \tag{4.27}$$

Constraints (4.26) and (4.27) replaces constraints (4.11) and (4.12) in the model.

4.3 Path Flow Model

The arc flow model was based on a model presented by Andersson et al. (2015). The path flow model can be seen as a reformulation of the arc flow model. A compact summary of the notation and model may be found in appendix B.

For the path flow model, the flow variables are replaced by path variables, that describe paths through the network of possible voyages (r,i). However, the speed weighting variables still has to be associated with each sailing leg, both voyage and ballast. In order to solve the problem, paths have to be generated a priori to solving the model. The path generation is discussed in further detail in Chapter 5. As discussed in Chapter 3.4, decomposition is a common solution method for solving large fleet deployment problems. In this case, the path flow model presented below can be considered as the master problem, and the a priori path generation for all ships can be considered as the subproblems. In this problem all ships have different properties, and thus no subproblems are identical and are thereby solved separately for each ship.

4.3.1 Additional notation for the Path Flow Model

Notation presented for the arc flow model is still valid for the path flow model. Therefore, only new notation for the path flow model is presented here. First of all, the paths need to be defined. Let $\mathcal{P}_v$ be a set of all feasible paths for ship v. $\mathcal{P}_{vriqj}$ is a subset of $\mathcal{P}_v$, which contains all paths where voyage j on trade route q directly follows voyage i on route r for ship v. $\mathcal{P}_{vri}$ is another subset of $\mathcal{P}_v$. This subset contains all paths where ship v sails voyage i on route r. The last subset of $\mathcal{P}_v$ that need to be defined is $\mathcal{P}_{vo(v)ri}$, containing all paths where voyage i on route r is the first voyage ship v performs from its origin.

Let E_{vpri} be a parameter that describes the earliest service start for ship v on voyage i on trade route r for a given path p.

Let z_{vp} be a binary variable, which equals 1 if ship v sail path p, and 0 otherwise. Let t_{vri} be a variable that sets the start time of voyage i on route r for ship v. The variable t^S_{ri} describes when a spot ship starts sailing voyage i on route r.

4.3.2 The Model

Objective function

$$\begin{aligned}
&\max \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{s\in\mathcal{S}}(R_{ri}-C_{vris})w_{vris} \\
&-\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{s\in\mathcal{S}} C^B_{vo(v)ris}w_{vo(v)ris} \\
&-\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}\sum_{s\in\mathcal{S}} C^B_{vriqjs}w^B_{vriqjs} + \sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{I}_r}(R_{ri}-C^S_{ri})u^S_{ri}
\end{aligned} \tag{4.28}$$

The objective function, (4.28), in the path flow model aims to maximize profit, by summing the most profitable combination of paths for the fleet and the most favorable spot ship options. This objective function is actually the same as for the arc flow model.

Service constraints

$$\sum_{v\in\mathcal{V}_r}\sum_{p\in\mathcal{P}_{vri}} z_{vp} + u^S_{ri} = 1, \quad (r,i)\in\mathcal{N}^C \tag{4.29}$$

$$\sum_{v\in\mathcal{V}_r}\sum_{p\in\mathcal{P}_{vri}} z_{vp} \leq 1, \quad (r,i)\in\mathcal{N}^O \tag{4.30}$$

$$\sum_{p\in\mathcal{P}_v} z_{vp} = 1, \quad v\in\mathcal{V} \tag{4.31}$$

Constraints (4.29) ensure that all contractual voyages are carried out exactly once, either by a ship within the fleet or by a spot ship. Constraints (4.30) ensure that the optional voyages may be carried out at most once and only by a ship within the fleet. All ships have to be assigned to exactly one path, constraints (4.31) make sure of that.

Network flow constraints

$$\sum_{s \in \mathcal{S}} w_{vris} = \sum_{p \in \mathcal{P}_{vri}} z_{vp}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.32}$$

$$\sum_{s \in \mathcal{S}} w^B_{vo(v)ris} = \sum_{p \in \mathcal{P}_{vo(v)ri}} z_{vp}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.33}$$

$$\sum_{s \in \mathcal{S}} w^B_{vriqjs} = \sum_{p \in \mathcal{P}_{vriqj}} z_{vp}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, q \in \mathcal{R}_v, j \in \mathcal{I}_q \tag{4.34}$$

The network flow constraints from the arc flow model has already been handled by the path generation. However, the speed weighting variables for each voyage must be connected to the paths. In other words, one must ensure that the speed weighting variables for a ship on a path can only take non-zero values if the ship sails that given path. This must be taken care of for each of the different sailing types. Constraints (4.32), (4.33) and (4.34) take care of the sailing of voyages, initial ballast sailing and ballast sailing, respectively.

Time constraints

$$\sum_{p \in \mathcal{P}_{vri}} E_{vpri} z_{vp} \leq t_{vri} \leq \sum_{p \in \mathcal{P}_{vri}} L_{ri} z_{vp}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.35}$$

$$E_{ri} u^S_{ri} \leq t^S_{ri} \leq L_{ri} u^S_{ri}, \quad r \in \mathcal{R}, i \in \mathcal{I}_r \tag{4.36}$$

$$\sum_{s \in \mathcal{S}} \left(T^B_{vo(v)ris} + E_{o(v)}\right) w^B_{vo(v)ris} \leq t_{vri}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.37}$$

$$\begin{aligned} &t_{vri} + \sum_{s \in \mathcal{S}} \left(T_{vris} w_{vris} + T^B_{vriqjs} w^B_{vriqjs} + (L_{ri} + T_{vri,1}) w^B_{vriqjs}\right) \\ &- L_{ri} - T_{vri,1} - t_{vqj} \leq 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, q \in \mathcal{R}_v, j \in \mathcal{I}_q \end{aligned} \tag{4.38}$$

Constraints (4.35) state that the starting time for a voyage has to be within the time window for that given voyage, and at the same time ensure that the variable equals zero if the given ship does not serve that voyage. The same goes for the starting time of spot ships in constraints (4.36). Constraints (4.37) ensure that a ship cannot start a voyage before it has sailed ballast from its origin position to the starting point of the voyage. Likewise, constraints (4.38) ensure that a ship cannot start a voyage before it has completed the previous voyage and sailed ballast to the start of the next voyage. The parameter $T_{vri,1}$ in constraints (4.38) is the longest possible time ship v can sail voyage (r, i). This is due to the fact that the set of speed points is an ordered set from the lowest speed to the highest speed, and thus sailing with speed point 1 is the longest time that ship v can sail voyage (r, i).

Evenly spread constraints

$$B_r + \sum_{v \in \mathcal{V}} t_{vri} + t^S_{ri} - \sum_{v \in \mathcal{V}} t_{vr,i+1} - t^S_{r,i+1} \leq 0, \quad r \in \mathcal{R}, i \in \mathcal{I}_r \backslash \{n_r\} \tag{4.39}$$

Constraints (4.39) are evenly spread constraints of consecutive voyages on the same trade. These constraints are slightly altered compared to the arc flow model, as the starting time variables in the path flow model are ship dependent.

Binary and Non-negativity Constraints

$$z_{vp} \in \{0, 1\}, \quad v \in \mathcal{V}, p \in \mathcal{P}_v \tag{4.40}$$

$$u^S_{ri} \in \{0, 1\}, \quad r \in \mathcal{R}, i \in \mathcal{I}_r \tag{4.41}$$

$$w_{vris} \in [0, 1], \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, s \in \mathcal{S} \tag{4.42}$$

$$w^B_{vriqjs} \in [0, 1], \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, q \in \mathcal{R}_v, j \in \mathcal{I}_q, s \in \mathcal{S} \tag{4.43}$$

$$w^B_{vo(v)ris} \in [0, 1], \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, s \in \mathcal{S} \tag{4.44}$$

$$t^S_{ri} > 0, \quad r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.45}$$

$$t_{vri} > 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{4.46}$$

Constraints (4.40)-(4.46) define all variables as either binary, weighing or continuous.

4.4 Modelling Choices

This section intends to discuss the choices that were made when formulating the mathematical models. The section mostly concerns the path flow model, as the arc flow model originates from the project assignment and only has minor changes. The choices that were made when formulating the model were done to achieve an accurate, tight and at the same time efficient formulation.

The use of subsets is an important element in the path flow formulation, to ensure that the correct paths are summed over in the constraints. By introducing the subsets of the set of paths, P_v, there is also a very favourable side effect which limits the number of necessary variables that is generated. Introducing P_{vriqj}, P_{vri} and $P_{vo(v)ri}$ is a smart way to handle all the paths, and to ensure that the correct share of paths is used in the respective constraints. An alternative approach would be to use logical matrices to state which paths can take a specific ballast sailing, a specific voyage or a specific initial ballast sailing, However, these matrices would be huge, which would be inefficient in the solution process.

The notation of voyages on a trade route is handled in the same way for both models. An alternative approach in the path flow model could be to use a formulation where every path is divided into sailing legs, indexed by l. This means that some indices (r, i) would be replaced by (p, l). For example, the speed weighting variables $w_{vo(v)ris}$, w_{vris} and w_{vriqjs} could be transcribed into the variable w_{vpls}. The major disadvantage of this approach is that the model would have several constraints defined for all possible combinations of paths and legs $(v \in \mathcal{V}, p \in \mathcal{P}_v, l \in \mathcal{L}_p)$. This would cause a vast increase in the number of constraints compared to having constraints for all combinations of trade routes and voyages, $(v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r)$. This is caused by the fact that the number of trade routes and voyages are limited to a relatively small number, whereas the number of paths is vastly increasing when more trade routes and voyages are added (this is described in more detail in Chapter 5). In addition, there would be an increase in the number of variables generated, as the indices give a good indication of the number of variables generated. The product of $(p \cdot l)$ is much larger than $(r \cdot i)$.

One of the major drawbacks with integrating voyage separation into the path flow model regards how it makes interdependencies between paths. This is because the starting times for a voyage not only depend on its time window, but also on the starting times for the previous and next voyages on the trade. This is required to fulfill the voyage separation requirements. Thus, the starting times have to be set in the master problem. Since the starting times are highly affected by the chosen speeds, the speed weighting variables have to be handled in the master problem as well. If there had not been any voyage separation requirements, only speed optimization, the speed optimization could have been performed individually on each path in the subproblems. In that case, the time windows would be taken into account when generating the paths, and the starting times and speed weighting variables would be set based on the sailing sequence of each path. This would make the master problem into a general set partitioning problem over the set of paths. The decisions to make would be which path to cover each voyage, where all voyages have to be served. However, the voyage separation in the problem at hand forces all variables concerning speed and time up to the master problem. This means that the master problem of the path flow model with speed optimization and voyage separation contains more variables and is much more complex than a path flow model without speed optimization. This necessary modelling choice suggests that the improvement in solution quality in the path flow model compared to the arc flow model is lower, than it would have been without speed optimization.

Chapter 5

Path Generation

This chapter concerns the description of the a priori path generation that was introduced in Chapter 4.3. Section 5.1 gives an in-depth description of the path generation and section 5.2 introduces path reduction heuristics and presents some heuristic rules.

5.1 Path Generation in General

For the path flow model, all feasible paths are enumerated a priori and then sent to the master problem. The path generation can be seen as a subproblem for each of the ships in the fleet, and all of the ships has to be treated individually as all ships have different properties. Each subproblem for each ship has to take time windows, compatibility and routing sequence into account. The ship starts at its initial position and seeks to find its first voyage to serve. The voyage has to be within reach, i.e. the ship has to arrive at the origin of the voyage within the time window (or earlier and wait until the window opens). Further, the ship has to be compatible with the trade the voyage is sailing, that is, the trade has to be a member of the set $\mathcal{R}_v$. All voyages that complies with these requirements will then be a possible first voyage and basis for the following path extensions. All possible initial voyages are further explored in the same way; in order to take on a consecutive voyage, the ship has to fulfill its ongoing voyage and sail ballast to the starting point of the new voyage. Here, it has to arrive within the time window (or earlier) and fulfill the compatibility requirements. An alternative to take on a new voyage is to end the path after the ongoing voyage. The ship has the possibility of staying idle through the planning period, as well. In the path generation process, maximum sailing speed is used when calculating the sailing duration on each leg (both ballast and voyage sailing). Using the maximum speed ensures that all feasible solutions are included in the solutions space of the problem, as proven by Norstad et al. (2011). This ensures that the path flow model has the same solution space as the arc flow model, which makes the two models comparable.

For each ship, this subproblem can be solved similar to a "breadth first" search tree algorithm. The search tree consists of nodes and arcs, where the voyages are represented as nodes and ballast sailing as arcs, respectively. The possible paths for one ship represented as a search tree is illustrated in Figure 5.1. The number of each node corresponds to the order of when the nodes are explored. This Figure corresponds to the path generation of all possible paths in the example in Section 2.4.

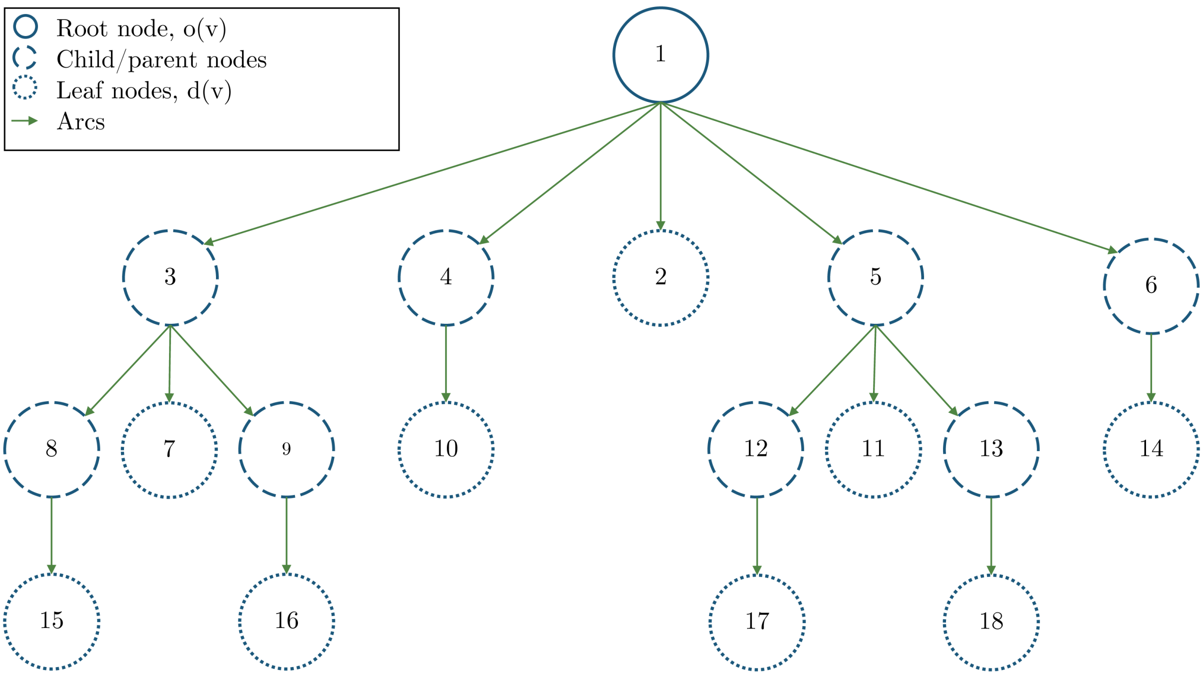

Figure 5.1: Breath first search tree

The root node in the search tree represents the origin node ($o(v)$), the initial position of the ship. All suitable initial voyages become child nodes of the root node, and the initial ballast sailing is represented as the initial arcs. All child nodes may then be explored, until all nodes have been branched and no more feasible voyages are possible to add to any path. The ship may end after any voyage, and all paths have to end at the artificial ending $d(v)$. This means that all nodes have to branch directly to $d(v)$, including directly from the root node. All the $d(v)$ nodes represents the leaves of the tree algorithm. Each path is characterized by a path number. The voyages on each path is not enumerated, but the parameter E_{vpri}, ensures that the voyages are carried out in the right order. E_{vpri} is a parameter that states the earliest starting time of voyage i on trade route r for ship v, on the given path p. The E_{vpri} is set as the latest time out of two options. Either, as the minimum time to execute all previous sailing legs on the path, including initial, voyage and ballast sailing, or the earliest time of the time window for the given voyage. This ensures that a subsequent voyage always has a higher E_{vpri} than the preceding voyages, which controls that all voyages are performed in the right order for each path. The path generator does not explicitly exclude the possibility that a voyage is included twice in the same path. However, there is assumed that no time windows are wide enough to allow a ship to sail the voyage, the ballast sailing back again, and start the same voyage for the second time within the time limits. Below, a pseudo code for the path generator is presented in Algorithm 1.

Algorithm 1 Pseudo code for the path generator

procedure PATH GENERATOR
    for $v \in \mathcal{V}$ **do**
        Branch directly from o(v) to d(v)
        for $r \in \mathcal{R}_v, i \in \mathcal{N}_r$ **do**
            if $E_{o(v)} + T^B_{vo(v)ris} \leq L_{ri}$ **then**
                Add (r, i) *as a child to the root node, generate path p*
                if $E_{o(v)} + T^B_{vo(v)ris} \leq E_{ri}$ **then**
                    $E_{pvri} = E_{ri}$
                else $E_{pvri} = E_{o(v)} + T^B_{vo(v)ris}$
        Develop tree from each node (r, i)*, given that* (r, i) *is a undeveloped child node*
        Branch directly from (r, i) *to* $d(v)$
        for $r \in \mathcal{R}_v, i \in \mathcal{N}_r$ **do**
            for $q \in \mathcal{R}_v, j \in \mathcal{N}_r$ **do**
                if $E_{pvri} + T_{vris} + T^B_{vriqjs} \leq L_{ri}$ **then**
                    Add (q, j) *as a child to the node (r,i), generate a new path p*
                    if $E_{pvri} + T_{vris} + T^B_{vriqjs} \leq E_{qj}$ **then**
                        $E_{pvqj} = E_{qj}$
                    else $E_{pvqj} = E_{pvri} + T_{vris} + T^B_{vriqjs}$
                    Branch directly to $d(v)$
    goto *top*

In path generation, there would be favourable to use a labeling algorithm and seek dominance criteria between paths, in order to reduce the total number of paths sent in to the master problem from the subproblems. This will in turn reduce both the generation time and the solution time for the solver. A path dominates another if it is at least as good or better on all criteria. However, due to the voyage separation requirements, which are handled in the master problem, not the subproblems, there is not possible to use dominance, and all suitable paths has to be included.

5.2 Path Reduction Heuristics

A major issue with the a priori path generation is the high number of paths that are generated compared to the number of ships, especially for the larger instances. Paths causes a large number of variables in the problem for the solver, which in turn impacts both solution time and quality. As a consequence, the solver struggles to handle the instances and obtains poorer solutions with a relatively high gap. For the path flow model, the optimal solution found by the solver uses one path per ship only. This means that if 100,000 paths are created, and there are 18 ships in the problem, only 18 paths out of 100,000 are used in the optimal solution. Thus, there are 99,982 paths that are generated, but not needed. Out of all paths that are not to be used, a significant share is unfavourable and likely not to ever be part of any optimal solution. The aim of heuristic rules is to reduce as many paths as possible, while keeping the loss in the profit to a minimum. For some problems, applying heuristics may achieve even better solutions than without. This happens if the problem is too large to solve to optimality, i.e. the original solution has a gap. The reduction in problem size makes the solution process easier for the solver, and a better solution is obtained. By applying heuristic rules, there is also a possibility to get solutions for problems of huge magnitude that are unsolvable otherwise. To eliminate paths that are most likely not included in the optimal solution, heuristic rules are implemented in the a priori path generation. These rules have some kind of acceptance criteria, and different levels of acceptance are tested in the computational study in Chapter 7. It is important to remember that when using heuristics, there is no guarantee of finding the actual optimal solution for the original problem. The trade-off between lack of optimality and reduced solution times are discussed in the computational study.

Four different kinds of a priori path generation heuristics have been implemented: Maximum percentage ballast sailing, maximum length of ballast sailing, maximum consecutive waiting days and minimum number of voyages per path. Each one is described in detail in the following sections. These rules may be applied separately or in combinations. The desired effect from combining heuristics is an even higher reduction in number of paths, and at the same time keeps the loss in profits to a minimum.

5.2.1 Maximum Percentage Ballast Sailing

As mentioned earlier, ballast sailing is highly undesirable for a shipping company due to high costs, and should be reduced to a minimum, even though some ballast sailing is a virtue of necessity. Thus, paths that contain a lot of ballast sailing are not desirable. This heuristic eliminates paths that contain more than a predetermined percentage of ballast sailing. For example, if the acceptance level is set to 35%, only paths that contain less than 35% ballast sailing are generated. The percentage is taken as total distance ballast sailing versus total sailing distance for the given path.

5.2.2 Maximum Length of Ballast Sailing

Another way of limiting the amount of ballast sailing is to eliminate all paths that contain legs with very long ballast sailing. The heuristic makes it impossible to sail ballast between two voyages where the ballast sailing distance exceeds the desired acceptance limit. For example, if the acceptance level is set to 10,000 nautical miles, the ability to sail ballast between voyages that have a higher ballast sailing distance is removed, thereby eliminating all paths with longer ballast sailing than 10,000 nautical miles. Again, one should be careful when exercising this heuristic, as long distance ballast sailing can be a part of an otherwise highly desirable path. It may also affect the solution space quite severe if the acceptance criteria is set too tight, and by that influence the solvability of the problem.

5.2.3 Maximum Consecutive Waiting Days

Waiting typically occurs when a ship arrives at the starting point of a voyage before the start of the time window, and has to wait before the voyage can be serviced. As speed optimization is considered, the amount of waiting on a path varies significantly from generation based on maximum speed, to the actual fleet deployment when speed has been adjusted. Thus, it is not easy to create a heuristic that consider waiting. However, one can use a heuristic rule that cuts all paths that have more than a certain amount of days of consecutive waiting. This ensures that paths with minor waiting, which most likely will be eliminated by speed optimization, passes through, while paths with long waiting periods are eliminated. Keep in mind that what looks like shorter waiting periods in the path generation, would most of the time not exist in the final solution. To remove all of these "waiting gaps" would reduce the possibility of speed reduction, i.e. the profit would decrease. Therefore, the acceptance criteria should not be fixed too low. This heuristic rule works as follows; an acceptance levels is set, and the path is disregarded if the waiting time exceeds this level during any point of the generation of the path. For example, if the acceptance level is set to 30 days, any path with waiting time higher than 30 days before starting a voyage is not created. The path generator is created in a way that any ballast sailing between two voyages is carried out straight after the first voyage. Hence, all potential waiting time between two voyages will accumulate ahead of the second voyage. As the waiting time that is eliminated is based on the highest speed it will also have the highest possible waiting time. Thus, this heuristic could be somewhat aggressive in the way it cuts paths, and the acceptance levels should be set fairly high and chosen with caution.

5.2.4 Minimum Number of Voyages per Path

In general, high fleet utilization is favorable for the shipping companies. Especially for longer planning horizons, this means that each ship has to execute several voyages in sequence within the period. In other words, paths with few voyages are undesirable for long planning horizons. Thus, another heuristic rule is to remove all paths that contain less than a given number of voyages. For example, if the acceptance level is set to three voyages, all paths that contain only one or two voyages are removed. Please note that this rule will only be used for large instances with long planning horizons (150 and 180 days), as instances with shorter planning horizons might need less voyages on a path to achieve a desirable solution.

Chapter 6

Test Instances

The purpose of this chapter is to present an overview of the data and test instances that are used to perform computational testing of the models presented in Chapter 4.

All tests in the computational studies are based on the same instances. The instances are based on data from Saga Forrest Carriers, the predecessor of the Norwegian shipping company Saga Welco AS. The instances are set up such that there is a well spread range of ships, trade routes, voyages and planning horizons. The size of each instance is most of all a function of the total number of voyages to execute. The number of voyages can be increased in two ways; increase the number of trade routes or the length of the planning horizon. Each trade route has an average frequency of voyages per month, which varies from one voyage per month up to four voyages per month. Hence, the number of voyages for an instance is a function of trade routes (and their respective frequencies) and length of the planning horizon.

The length of the planning horizon resembles how many voyages are to be executed in series. Whereas, the number of trade routes resembles how many voyages are to be executed in parallel. Hence, the number of ships required are dependent on the number of trade routes for an instance, not the planning horizon. The number of ships in each instance should be decided carefully. Not only the number of trade routes, but also the frequency per trade and the distance per voyage determines the required fleet size to serve the instance. This has been considered when creating instances. The instances are divided in four sets, one large set, one small and two in between. The instances have either six ships and three trade routes, ten ships and five trade routes, fourteen ships and seven trade routes or eighteen ships and nine trade routes. For each set, there are 3 planning horizons, 60, 90 or 120 days. The first three instance sets are all reduced variants of the fourth instance set, where some trade routes and ships have been removed.

Fuel costs have been set to 388 \$/ton. This value was the global average for the 20 largest ports in the world for the first quarter of 2018 (Ship & Bunker, nd). However, one should keep in mind that fuel price may vary a lot with time, and it may change rapidly. As described in 4.1, the fuel consumption function is typically a convex function. The instances use a second order polynomial (in terms of tons per nautical mile) to calculate the fuel consumption. This polynomial is: $Q(h) = A \cdot h^2 + B \cdot h + C$. The fuel coefficients were not provided in the data from Saga Forrest Carriers, but reasonable values were provided separately by the supervisors. The same fuel coefficients, and thereby the same fuel consumption curve, is applied for all ships and voyages. The coefficients used are $A = 0.0006$, $B = -0.0111$ and $C = 0.1411$. This fuel consumption function is applied when the ships are fully loaded. In order to adjust the fuel consumption of ballast sailing, the fuel consumption is multiplied by a factor of 0.8. Minimum and maximum speed for each ship are provided by the Saga Forrest Carriers data. In this study, all ships have the same speed range.

Table 6.1 describes the instances used in the computational study in detail. The "Instance" column names all test instances used, from 1 to 12. Later on, this number is refereed to when identifying the instance of interest. The "Set" column divide the instances into sets based on the number of ships in the instance. The "Voyages" column describes how many voyages that should be carried out for each instance. The numbers represent the contractual voyages out of the total number of contractual and optional voyages. As mentioned in Section 2.1, Figure 2.1 illustrates the trade routes used in the test instances. The trade routes are numbered according to appearance. For example, the instances with 3 trade routes consist of trade routes 1-3, instances with 5 trade routes consist of 1-5, and so on. In this study, the optional voyages are organized as one trade that consists of optional voyages only. As Figure 2.1 illustrates, trade route 5 represents the optional voyages.

Table 6.1: Overview of test instance specifications

Set	Instance	Ships	Trades	Voyages	Horizon
1	1	6	3	11/11	60
	2	6	3	15/15	90
	3	6	3	20/20	120
2	4	10	5	13/15	60
	5	10	5	18/21	90
	6	10	5	24/28	120
3	7	14	7	24/26	60
	8	14	7	34/37	90
	9	14	7	46/50	120
4	10	18	9	30/32	60
	11	18	9	44/47	90
	12	18	9	59/63	120

For the instances in Table 6.1, all feasible paths are calculated a priori, and the total number of paths for each instance are shown in Table 6.2 below. The path generation is performed separately from the optimization solver. The calculation time of path generation is not included in the run times of the path flow model (shown in Table 7.1). Relative to the running times of the path flow model itself, the path generation times are quite small, and are therefore neglected. The generation time may be pushed further down by perfecting the code of the path generator and use a more adequate software and programming language, e.g. C++. This is however beside the scope of this thesis. The number of paths increase vastly when the instance size increases. Within each set (1-4), the increase in number of paths has an exponential trend, as the time horizons are extended. This is due to the fact that longer time horizons increase the options for paths with several voyages in sequence.

Table 6.2: Number of paths per instance

Instance	Paths
1	159
2	299
3	985
4	364
5	823
6	3,277
7	1,886
8	8,711
9	69,776
10	3,073
11	16,199
12	138,292

Table 6.3 presents the perfect spread and the minimum number of spread (B_r) per trade route. All values are in number of days. Perfect spread is the ideal number of days between two consecutive voyages on a given trade. Note that trades with the same frequency, i.e. number of days between voyages with perfect separation, does not necessarily have the same values for minimum spread.

Table 6.3: Minimum spread of days between consecutive voyages per trade

Trade	Perfect spread (days)	Min. spread (days)
1	30	20
2	30	15
3	30	15
4	7	5
5	10	5
6	30	10
7	30	15
8	14	8
9	30	0

All instances of our mathematical programming model have been solved with a Windows 10 computer with an i7-7700 quad-core 3.60 GHz CPU, 32 GB DDR4 RAM and a 512 GB PCIe SSD drive. The software used is Fico Xpress IVE 64-bit with optimizer version 31.01.09. The path generation algorithm has been implemented in MATLAB R2017a.

Chapter 7

Computational Study

In this chapter, the results of the computational study are presented. First, a comparison of the arc flow and path flow models are discussed in Section 7.1, followed by an analysis of different number of speed points, and the effect of speed optimization in Section 7.2. After that, results of applying heuristic rules are presented in Section 7.3. The results when combinations of the heuristic rules are applied on larger instances are presented in Section 7.4. Section 7.5 presents a study of the effect of voyage separation. The chapter is rounded up with a discussion of the results in Section 7.6.

7.1 Comparison of the Arc Flow and Path Flow Models

This section compares the performance of the arc flow and path flow models, with respect to objective value and solution time. All instances presented in Table 6.1 have been run for both the arc flow and path flow models. Both models have been run with 3 speed points (minimum, average and maximum), which gives an estimation of the fuel consumption similar to the one illustrated in Figure 4.1. All profits and objective values described in this thesis are in USD. If an instance is solved to optimality, it is proven that no other fleet deployment can increase the profit further. If an instance is not solved to optimality within the time limit, the objective value presented is the current best solution. The best solution found represent a lower bound for the optimal solution, i.e. the optimal solution is at least as good, or better than the current solution. The gap between the lower and upper bound, is called the optimality gap, and is presented in percent. Figure 7.1 illustrates the relations between the current solution, optimal solution, best bound and LP relaxation of the problem. The values are in increasing order, as the models in this thesis are formulated as maximization problems.

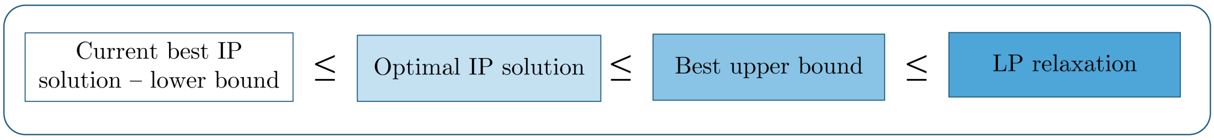

Figure 7.1: Relation between current IP solution, optimal IP solution, best bound and LP relaxation.

The results of the tests are compared in Table 7.1. The ”Time” columns indicate the computational times in seconds. The models have all been run with an upper time limit of 3600 seconds (1 hour). The ”Obj val.” columns report the best objective value, the profit, obtained by the current best fleet deployment. The optimality gap is shown in the ”Gap” columns. The solution of the LP relaxations of the problems are shown in the ”LP Rel.” columns.

Table 7.1: Comparison of arc flow and path flow models for 3 speed points.

	Arc Flow Model				Path Flow Model			
Instance	Time	LP Rel.	Gap	Obj val.	Time	LP Rel.	Gap	Obj. val.
1	0.6	14,742’	0.00%	13,837’	0.1	13,949’	0.00%	13,837’
2	1.7	19,874’	0.00%	17,350’	1.2	17,965’	0.00%	17,350’
3	80.6	26,123’	0.00%	22,223’	10.6	23,308’	0.00%	22,223’
4	1.3	18,035’	0.00%	17,456’	0.3	17,555’	0.00%	17,456’
5	1857.1	24,485’	0.00%	22,949’	13.0	23,845’	0.00%	22,949’
6	3600.0	32,018’	11.77%	28,141’	3600.0	31,090’	3.99%	28,795’
7	3600.0	26,186’	4.72%	24,995’	1704.3	25,835’	0.00%	25,339’
8	3600.0	37,752’	18.62%	31,579’	3600.0	35,967’	4.56%	33,934’
9	3600.0	50,587’	23.10%	40,751’	3600.0	47,621’	12.62%	42,227’
10	3600.0	31,610’	5.20%	29,435’	3600.0	30,621’	0.14%	30,288’
11	3600.0	44,466’	28.61%	34,419’	3600.0	43,465’	6.43%	40,755’
12	3600.0	59,340’	114.99%	27,510’	3600.0	58,223’	13.76%	51,161’
Average	2261.8	32,101’	17.25%	25,887’	1944.1	30,787’	3.46%	28,860’

The results in Table 7.1 show that the small instances (1,2 and 4) are solved to optimality by both models, and there are marginal differences in solution time. However, for the larger instances there are significant differences between the two models, in favor of the path flow model. The difference is especially noticeable for instance 7, where the path flow model finds the optimal solution in less than half the time the arc flow model. The arc flow model does not obtain optimality, the solution has a 4.72% gap. The fact that the path flow model indeed has an overall, better performance than the arc flow model, becomes clear when comparing the average values. The path flow models achieve solutions with objective values 11.46% better than the arc flow model on average. With regards to the solution times, the path flow model is on average 14.04% better. A major reason for the better performance of the path flow model is the results of the LP relaxations, which are considerable tighter for the path flow model than for the arc flow model. This gives a lower initial upper bound, and the optimality gap are confined faster. The path flow model achieves LP relaxations that are on average 4.08% better than the arc flow model. On average the arc flow model and path flow model achieve gaps of 17.25% and 3.46%, respectively. As a consequence of these results, only the path flow model is evaluated further.

7.2 Comparison of Different Speed Points

This section contains a comparison of the effect of speed optimization, and how the number of speed points affects solution quality. To perform such a comparison, the path flow model has been run for one, two and three speed points, respectively. The instances used are the ones with a planning horizon of 120 days (see Table 6.1). The results from these comparisons are shown in Table 7.2. The profits in the "Profit" columns are a posteriori values. The a posteriori speed optimization has been solved by using ten discrete speed points, which gives negligible deviations from true fuel consumption function for this case. The a posteriori optimization ensures that profit for all instances are compared on the same basis. As a consequence of the a posteriori values, the values in the "Profit" column for three speed points deviate slightly from the "Obj val."-values in Table 7.1. The "Gap" columns indicate the optimality gaps obtained before the a posteriori optimization. The "Time" columns report the computational times in seconds.

Table 7.2: Comparison of different number of speed points.

	1 speed point (max)			2 speed points (max/min)			3 speed points (max/avg/min)		
Instance	Time	Gap	Profit	Time	Gap	Profit	Time	Gap	Profit
3	0.1	0.00%	20,175'	11.1	0.00%	22,246'	10.6	0.00%	23,308'
6	0.2	0.00%	27,666'	3600.0	5.56%	28,555'	3600.0	3.99%	28,852'
9	211.4	0.00%	41,207'	3600.0	13.61%	42,367'	3600.0	12.62%	42,359'
12	994.3	0.00%	51,173'	3600.0	16.16%	50,781'	3600.0	13.76%	51,553'
Average	301.5	0.00%	35,055'	2702.8	8.83%	35,987'	2702.7	7.59%	36,518'

When reviewing Table 7.2, the instances with one speed point are all solved to optimality within the maximum time limit. For both two and three speed points, three of four instances are run for 3600 seconds, the maximum time. However, the profits are lower than for the instances with speed optimization. Two and three speed points achieve profits that are 2.66% and 4.17% better than without speed optimization, respectively. This means that there is a trade-off between solution time and the obtained profit, when discussing which model performs best. Based on the fact that the solution time is within an hour, whereas profit is in terms of million dollars, the models with integrated speed optimization is considered to have a higher utility value. When comparing the results for two and three speed points, the average solution times are essentially the same, but the profit obtained is somewhat better when using three speed points. Therefore, all further model evaluations are performed with three speed points. When comparing the a posteriori profit values in table 7.2, and the objective value of the path flow formulation in Table 7.1, there are fairly small differences. This in turn shows that a linearization based on linear combinations of three speed points gives a good approximation of the real profit. Bear in mind that the a posteriori profit will always be at least as good as the obtained objective value. For convenience, the word "profit" will throughout this thesis be used to describe the profit obtained as objective value, but without a posteriori speed optimization.

Another observation from running the models for different number of speed points, is the number of voyages performed by spot ships. The use of spot ships per instance is shown in Table 7.3.

Table 7.3: The use of spot ships for the models with different number of speed points.

Instance	1 speed point	2 speed points	3 speed points
3	3	3	3
6	1	3	2
9	0	7	6
12	0	1	3
Average	1	4	4

The number of voyages carried out by spots ships are higher when integrating speed optimization into the model. The model with one speed point charter in spot ship only to serve one voyage (on average). Both of the models with two and three speed points charter in spot ships to serve four voyages (on average). All values are average number of voyages. Especially instance 9 raises the average, where seven and six voyages are served by spot ships for two and three speed points, respectively. A likely reason for this could be that instance 9 is a bit unbalanced regarding the amount of sailing to carry out, and the number of ships. Nonetheless, none of the instances uses an unreasonable number of spot ships. It should also be pointed out that all the instances are solved to optimality for the model with one speed point, where as there are optimality gaps for the instances for the models with two and three speed points (except the smallest one). A proven optimal solution may use fewer spot ships. The use of some spot ships gives more planning flexibility and may be profitable to use, hence the use of spot ships is not necessarily negative.

7.3 Path Reduction Heuristics

This section regards a study of the path reduction heuristics presented in Chapter 5.2. In this section, the effectiveness of each heuristic rule is tested. The main goal when implementing heuristics is to achieve fairly good solutions faster, or to be able to solve larger instances that are unsolvable (within reasonable time) without any heuristics. Therefore, the largest instance within each set 2-4 (instance 6, 9, 12) are used during the testing of the heuristics. All of these instances have obtained fairly good solutions within the solution time of 3600 seconds, as shown in Table 7.1, but not optimal. The profits without heuristics are used as reference values throughout this section, in order to measure the performance of the heuristic rules. When considering path reduction heuristics, there are two effects that influence the solution quality and they work in the opposite directions. When removing paths, there is a risk that some good, or even optimal, paths are eliminated from the problem. That does in turn cause the heuristics to yield poorer solutions since good paths are removed from the solution space. At the same time, the reduction in paths results in a smaller, less complicated problem for the solver, and thus the solver is able to get closer to the true optimal solution of the (reduced) problem and obtain a better solution. Hence, as long as the original solution is not the optimal solution, a heuristic may obtain solutions with either increasing or decreasing objective values compared to the original solution. If the original problem has been solved to optimality, the solutions obtained with heuristics applied, can never surpass the original solution.

7.3.1 Maximum Percentage Ballast Sailing

Results for running instances with the maximum percentage ballast sailing heuristic rule is given in Table 7.4. The heuristic is tested for four carefully selected acceptance levels, 45%, 40%, 35% and 30%. These values represent the maximum percentage ballast sailing compared to total distance sailed in each path. The values of the acceptance levels are selected based on the case data and preliminary testing, and ensures a fair variation in number of paths removed, without compromising the profit too much.

Table 7.4: Results for the maximum percentage ballast heuristic.

Instance	% Max. Ballast	Paths	% Paths Eliminated	Time	Profit	% Impr,
6	-	3,277	0%	3600.0	28,795'	-
	45%	2,076	36%	3600.0	28,821'	0.09%
	40%	1,729	46%	3600.0	28,816'	0.07%
	35%	1,160	64%	31.2	28,531'	-0.92%
	30%	833	74%	8.8	27,506'	-4.47%
9	-	69,776	0%	3600.0	42,227'	-
	45%	35,002	50%	3600.0	42,290'	0.15%
	40%	22,006	68%	3600.0	43,053'	1.95%
	35%	10,840	84%	3600.0	43,480'	2.97%
	30%	6,623	91%	3600.0	43,732'	3.56%
12	-	138,292	0%	3600.0	51,161'	-
	45%	75,604	45%	3600.0	48,826'	-4.56%
	40%	50,989	63%	3600.0	51,457'	0.58%
	35%	33,160	76%	3600.0	51,154'	-0.01%
	30%	19,396	86%	3600.0	51,906'	1.46%

When analyzing the results in Table 7.4, there are several interesting findings. First of all, the enormous reduction in run time for instance 6 for acceptance level 35% 30% should be noted. These instances have a reduction in run time above 99%, 99,13% and 99,76% to be exact. The test with acceptance level of 35 % yield a very good solution quality, whereas the one with 30 % acceptance level has a severe decrease. Throughout these results, 8 of the 12 tests achieve an even higher profit within the same run time, as the original instance did without the heuristic rule. Another interesting result here is for the 30% acceptance level. The heuristic cuts as much as 74%, 91% and 86% of all paths (average of 84%), while upholding a good profit. Based on the results it is evident that this heuristic rule is quite effective. They yield the desired effects of reduction in run time or increase in the solution quality, and at the same time reduces the number of paths drastically.

7.3.2 Maximum Length of Ballast Sailing

The results of running instances with the maximum length of ballast sailing heuristic rule are provided in Table 7.5. The heuristic is tested for two acceptance levels, 10,000 and 10,500 nautical miles. These values have been chosen based on the mean plus the standard deviation of ballast distance. The mean and standard deviation is calculated out of the set of the distances of all ballast sailing legs for each instance. On average, the sum of the mean and the standard deviation is between 10,000 and 10,500 nautical miles. The threshold values for this heuristic has therefore been rounded to these two values.

Table 7.5: Results for the maximum length of ballast sailing heuristic.

Instance	Max. Ballast Distance	Paths	% Paths Eliminated	Time	Profit	% Impr.
	-	3,227	0%	3600.0	28,795'	-
6	10,500	2,395	26%	51.2	28,451'	-1.20%
	10,000	1,659	49%	32.7	27,920'	-3.04%
	-	69,776	0%	3600.0	42,227'	-
9	10,500	46,618	33%	3600.0	42,274'	0.11%
	10,000	37,774	46%	3600.0	42,511'	0.67%
	-	138,292	0%	3600.0	51,161'	-
12	10,500	93,022	33%	3600.0	49,205'	-3.82%
	10,000	78,359	43%	3600.0	50,599'	-1.10%

The results of the maximum length of ballast sailing heuristics shows a greater impact on the solution quality, compared to the maximum percentage ballast sailing heuristics. Here, both instance 6 and 12 have a reduction in profit when applying the heuristics, where instance 9 has marginally better solutions. As for the previous heuristic, instance 6 has a vast reduction in run time, here as well. In the trade-off between solution time and quality, these values for instance 6 are promising. Even though this heuristic removes quite few paths, in the range 26 % - 49 %, the effect on the profit is more severe, compared to the previous one. This indicates that the acceptance criteria for this heuristic should be chosen with utmost care. The huge advantage by using this heuristic is that it excludes some ballast legs completely, both in the subproblems and in the master problem. This causes fewer variables, and less input data to handle, which is favourable.

7.3.3 Maximum Consecutive Waiting Days

Finally, for the individual testing of the heuristic rules, results for the maximum consecutive waiting days heuristic rule is given in Table 7.6. The heuristic is tested for three carefully selected acceptance levels, 30, 20, and 10 days. These levels are chosen based on preliminary testing of the instances.

Table 7.6: Results for the maximum consecutive waiting days heuristic.

Instance	Max. Waiting Days	Paths	% Paths Eliminated	Time	Profit	% Impr.
6	-	3,227	0%	3600.0	28,795'	-
	30	2,554	21%	3600.0	28,857'	0.21%
	20	2,077	36%	3600.0	28,957'	0.56%
	10	1,365	58%	3600.0	28,964'	0.59%
9	-	69,776	0%	3600.0	42,227'	-
	30	59,273	15%	3600.0	42,603'	0.89%
	20	46,681	33%	3600.0	42,693'	1.10%
	10	29,383	58%	3600.0	42,443'	0.51%
12	-	138,292	0%	3600.0	51,161'	-
	30	120,876	13%	3600.0	49,502'	-3.34%
	20	97,648	29%	3600.0	50,943'	-0.43%
	10	63,545	54%	3600.0	49,879'	-2.51%

The results of the maximum consecutive waiting days heuristic are quite stable. For instance 6, this is the only heuristics that consequently achieve better solutions. These are good results regarding the fact that instance 6 has a low optimality gap without heuristics (only 3,99 %). Instance 9 gets promising results as well, whereas instance 12 has a small decrease. Relatively to the other instances, instance 12 has the lowest reduction in number of paths (in percent). The percentage of paths removed varies from 13% up to 58%.

7.4 Combinatorial Heuristic Study

This section shows the results from the combinatorial heuristic study. The study of combining the different heuristics is based on the results from running the heuristics isolated. The combinations are formed by combining the various heuristic rules from the previous section for different acceptance levels.

7.4.1 Testing of Heuristics Combinations

As shown in the previous subsection, the chosen values for each heuristic rule have provided various objective values by themselves without extreme cuts in number of paths. Now, the aim is to maintain relatively good solution quality, but to remove a much higher number of paths. Since the different heuristic rules has different functioning mode, they are likely to remove different paths from the path set. All heuristic combinations are presented in Table C.1. Each combination has been named by a letter from A to X. These letters will be refereed to later on, when a specific combination is in mind. All possible combinations of heuristics have been tested and evaluated for instance 6, 9 and 12. By running the heuristic combinations on instances that are already solved, the solutions are comparable and gives information on the solution quality. Complete results are to be found in Table C.2. Table 7.7 shows the specifics of the combinations that are further analyzed throughout this section. Out of all the 24 combinations, combinations E, G and J performed best on average. In addition to the three highest performing combinations, combination X is also included. This combination is the most aggressive one and removes most paths.

Table 7.7: Heuristic combinations

Comb.	Max Length Ballast Sailing	Max. Consecutive Waiting Days	Max. % Ballast Sailing
E	10,500	20	45 %
G	10,500	20	35 %
J	10,500	10	40 %
X	10,000	10	30 %

The results of running the heuristic combinations from Table 7.7 are shown in Table 7.8.

Table 7.8: Results of heuristic combinations.

	Instance 6			Instance 9			Instance 12			Average	
Comb.	Paths	Profit	% Impr	Paths	Profit	% Impr.	Paths	Profit	% Impr.	Profit	% Impr.
None	3,277	28,926'	-	69,776	42,227'	-	138,392	51,161'	-	40,771'	-
E	970	28,327'	-1.63%	19,603	42,716'	1.16%	42,338	51,841'	1.33%	40,961'	0.29%
G	596	27,946'	-2.95%	8,127	43,075'	2.01%	20,431	52,098'	1.83%	41,040'	0.30%
J	487	28,240'	-1.93%	7,542	43,208'	2.32%	18,294	52,305'	2.24%	41,251'	0.88%
X	268	26,659'	-7.42%	2,365	42,113'	-0.27%	6,956	52,740'	3.09%	40,504'	-1.53%

Combinations E, G and J have a positive, average improvement, especially for the larger instances (8 and 12). Combination X performs exceptionally well for the largest instance (12). Combination X removes 95% of all paths, and still manages to get an improvement of 3,09%. The results for Instance 6 may seem quite poor as they all have a negative improvement, but the solutions are achieved much faster than without heuristics.

7.4.2 Results of Larger Instances

An important feature of heuristics is to solve instances that does not achieve any solutions within a reasonable amount of time by the original model. In order to test this effect, a new set of instances are used. These instances are based on instance 6, 9 and 12, respectively, but the time horizon have been extended up from 120, to 150 and 180 days. Due to the magnitude of the problem, instance 12 has only been extended to 150 days, not 180. An overview of these new instances is displayed in Table 7.9.

Table 7.9: Overview of specifications for larger test instance.

Instance	Ships	Trades	Voyages	Horizon
13	10	5	31/36	150
14	10	5	37/43	180
15	14	7	58/63	150
16	14	7	69/75	180
17	18	9	74/79	150

These new instances have been run without applying any heuristics or combinations of these. This is done out of two reasons, first, to see how they perform by themselves.

Second, to obtain a reference value for the heuristics, in order to say something about the solution quality and path reduction when heuristics are applied. The results of this testing are displayed in Table 7.10 below.

Table 7.10: Results of larger instances without heuristics.

Instance	Paths	Time	Gap	Profit
13	20,817	3600.0	9.77%	35,275'
14	83,491	3600.0	14.39%	40,318'
15	642,487	86400.0	14.15%	50,830'
16	6,220,000[1]	-	-	-
17	1,395,830	86400.0	18.70%	61,787'

[1] Estimated number of paths

Table 7.10 shows that Instance 13 and 14 actually obtain fairly good solutions within 3600 seconds without any heuristics applied, even though the planning horizon has been extended. Instance 15 obtained its first integer solution after 6431 seconds, of 49,971' and an gap of 19,43 %. However, due to this relatively large gap, the instance was run for 24 hours in order to obtain a better solution. This to ensure a fair basis for comparison and measurement of the performance of the heuristics. As for Instance 17, it has been run for 24 hours as well. However, compared to Instance 15, the first solution was found much later, after 78209 seconds, and this solution were the only one obtained. For Instance 16, the instance is far too large for the path generator and the solver to handle, and no solution has been obtained at all. However, an estimation of total number of paths, based on the number of paths for some ships within the instance are presented.

When solving these larger instances, the heuristic combinations presented in Table 7.7 are utilized. Results from running Instance 15-17 are shown in Tables 7.11, 7.12 and 7.13, respectively.

Table 7.11: Results of instance 15

Comb.	Paths	Time	Profit	% Impr.
None	642,487	86400.0	50,830'	-
E	149,659	3600.0	51,043'	0.42%
G	52,939	3600.0	52,396'	3.08%
J	48,839	3600.0	51,766'	1.84%
X	12,261	3600.0	51,992'	2.29%

For instance 15, with 14 ships and 150 days horizon, the results with heuristic rules applied are quite good compared to the result without heuristics. The solutions obtained are consistently better than the original solution, and the solutions are obtained within 3600 seconds, which is a massive improvement from the original solution time.

Table 7.12: Results of instance 16

Comb.	Paths	Time	Gap	Profit
None	6,220,000	-	-	-
J	232,073	3600.0	22.01%	57,041'
X	48,024	3600.0	12.99%	59,035'

For instance 16, where the time horizon has been prolonged up to 180 days, the model does not achieve the desired outcomes. The "Gap" column here represents the gap of the problem with heuristics applied. The true optimality gap is at least as large as the gaps presented here, likely much larger. Within the third instance set (Instance 7-9), which instance 15 and 16 are extensions of, there is a trend of the profit increasing by approximately 8,500' USD per 30 days that are added to the planning horizon. This trend seems to be slightly increasing per 30 days that are added. Based on this, the profits for a planning horizon of 180 days should approximately be at least 60,000' USD. This is another indication that the obtained solutions are quite poor. Hence, the problem is of a magnitude where the path flow model with heuristics does not work well.

For instance 17, based on preliminary testing and lack of solutions within 3600 seconds, the maximum solution time has been increased up to 14400 seconds. In addition, the heuristic rule that require a minimum number of voyages per path is also tested. Results are displayed in Table 7.13, both with and without the minimum number of voyages per path heuristic rule. The limit for minimum number of voyages per path is set to four voyages. This last heuristic rule is only applied in combination with the heuristic combinations earlier presented, and not alone.

Table 7.13: Results of instance 17

	Instance 17				Instance 17 min. 4 voy/ship			
Comb.	Paths	Time	Profits	% Impr.	Paths	Time	Profits	% Impr.
None	1,395,830	86400.0	61,737'	-	-	-	-	-
E	343,809	14400.0	62,737'	1.54 %	311,596	14400.0	60,446'	-2.17 %
G	141,834	17347.0[1]	62,840'	1.70 %	125,945	14400.0	62,566'	1.26 %
J	126,319	14400.0	62,202'	0.67 %	115,972	14400.0	62,579'	1.28 %
X	38,354	14400.0	63,270'	2.40 %	34,299	14400.0	63,491'	2.76 %
Average	162,579	15136.8	62,762'	1.58 %	146,953	14400.0	62,271'	0.78 %

[1] First solution found

Table 7.13 show that Instance 17, an instance that barely achieved one solution within 24 hours, obtain reasonably good solutions within four hours when heuristic rules applied. With the exception of combination E with minimum 4 voyages, all combinations obtain higher profits. This may be explained by the fact that the relatively large number of paths, and that all ships have a minimum number of voyages that they need to sail, makes the problem relatively hard to solve. An outstanding result for this instance, is the result of the fiercest heuristic combination, Combination X with minimum 4 voyages. This heuristic obtains the highest profit of all combinations. The combination eliminates 98 % of all possible paths, and the solution has an improvement of 2,76 %. Compared to the best bound of the original run of instance 17, the optimality gap of Combination X with minimum four voyages is 15,51 %. Solutions of good quality for an instance that originally had almost 1.4 million paths are obtained within four hours. The effect of path reduction heuristics is here proven to be an effective tool in order to solve larger problem instances within reasonable time and with good solution quality.

7.5 Voyage Separation Requirement

This section regards a study of the effect of the voyage separation requirement. In the first part of this section, a study that considers the relationship between the objective value and the level of voyage separation is performed. The second part looks into the effect of the combination of voyage separation and speed optimization, where a specific trade is used as an example.

7.5.1 Relation Between Objective and Voyage Separation

It is interesting to see what effect the voyage separation requirement has on the objective value of the model. One way of analyzing this effect would be to test different relaxations and restrictions of the voyage separation requirement. In other words, one would like to scale the separation parameter (B_r), and observe how the objective value is affected by different values of B_r. Figure 7.2 shows how the voyage separation affects the objective value based on Instance 8. In the figure, B_r is scaled in a range from 0 to 1.75, where 1 equals the initial B_r-values, and 0 is the equivalent of none voyage separation requirements at all. This range ensures that all possible objective values are estimated, as a scaling of 1.75 makes the problem infeasible. This is due to the combination a very high B_r-value and the time windows. The value of B_r when the problem turns infeasible varies from instance to instance.

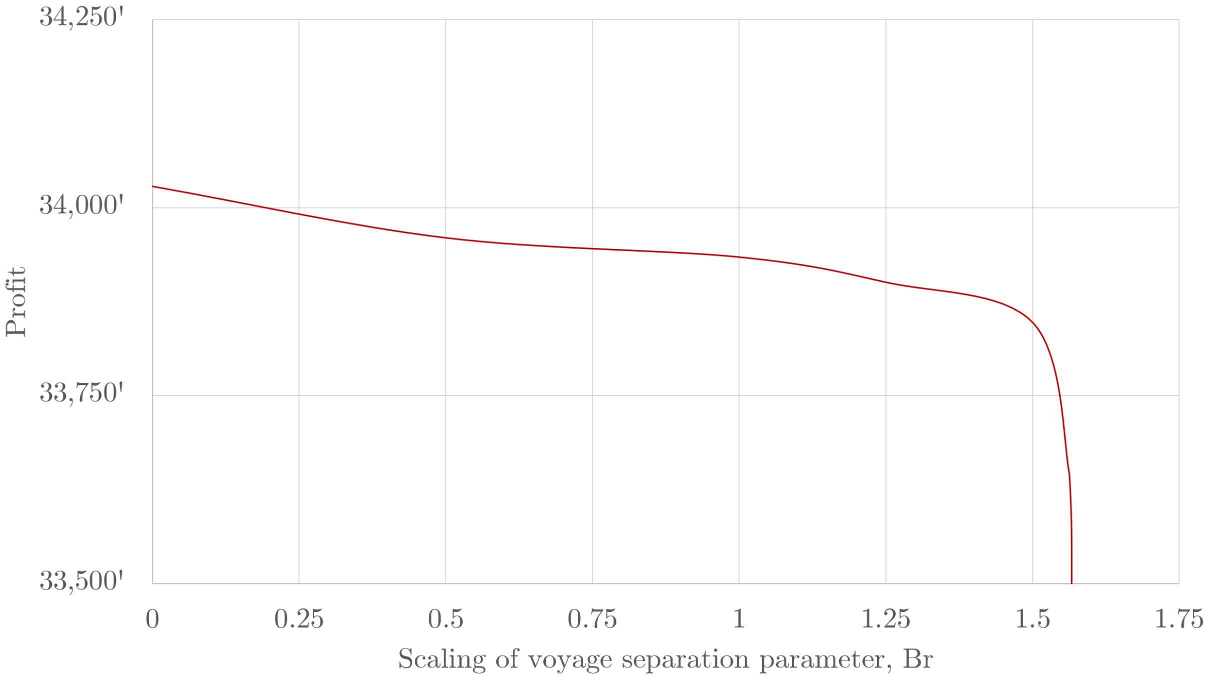

Figure 7.2: Profit as a function of the voyage separation parameter, B_r

Figure 7.2 shows that a higher relaxation of the voyage separation requirement gives a marginally better objective value. This behaviour is as expected. However, the solutions with low values for B_r are most likely undesirable as they probably would breach the "fairly evenly spread" term of the CoAs. The level of the voyage separation requirement does not seem to have a very high impact on the solution value, as it is only for very high values of B_r that there is a significant drop in objective value. This proves that adding the voyage separation requirement to the model causes a very low reduction in objective value, and increases the delivered service quality to the charterers. The terms of the CoAs are adhered to.

7.5.2 Relation Between Voyage Separation and Speed Optimization

As this thesis concerns both the voyage separation requirement and speed optimization, it is interesting to see how these two components behave in relation to each other. The way that this is done is by checking how the voyage separation on one specific trade route is affected by turning on and off the voyage separation and speed optimization.

The possible combinations of voyage separation and speed optimization are: with both speed optimization and voyage separation, with speed optimization and without voyage separation, without speed optimization and with voyage separation and without both speed optimization and voyage separation.

The voyage separation is compared by calculating the standard deviation of the intervals between the actual starting times, compared to the perfect spread between voyages. The trade route that is checked is the one with the most voyages. This trade route has a perfect spread of 7 days, as the average frequency is 7 days (the number of days between two following time windows are opened). The time windows for each voyage spans over 10 days. Thus, the time windows for the voyages on this given trade route are overlapping. The results from comparing combinations of speed optimization and voyage separation for this given trade are presented in Table 7.14. In order to make the results more reliable, the standard deviations presented are average standard deviations, based on the actual starting time for the voyages on the given trade, from several instances.

Table 7.14: Effects of voyage separation and speed optimization combined

	Without Voyage Separation	With Voyage Separation
With Speed Optimization		
Standard deviation in days	4.65	2.69
Standard deviation in %	66%	38%
Without Speed Optimization		
Standard deviation in days	3.87	2.81
Standard deviation in %	55%	40%

At first glance, Table 7.14 shows, as expected, that the separation of voyages is significantly better when the voyage separation requirement is applied. In addition, there are a couple more takeaways from this table that are important to point. When comparing the separation with and without speed optimization it is evident that the improvement in separation is higher when speed optimization is applied. This is an indication that it should be of even higher importance to include the voyage separation requirement when speed optimization is applied. Implementing speed optimization gives a higher probability of utilizing broader parts of the time windows. Hence, a greater variation in selected starting times for the voyages, and thus a worse separation when voyage separation is not considered. This is further substantiated by the fact that the worst separation is not achieved when neither of the two are applied, but when speed optimization alone is applied.

In order to illustrate the improved separation when the voyage separation requirement is applied, the starting times of one trade route has been illustrated in Figure 7.3 with and without voyage separation. The illustration is purely meant as an example of how the spread of voyages on a trade route can be affected by the voyage separation requirement.

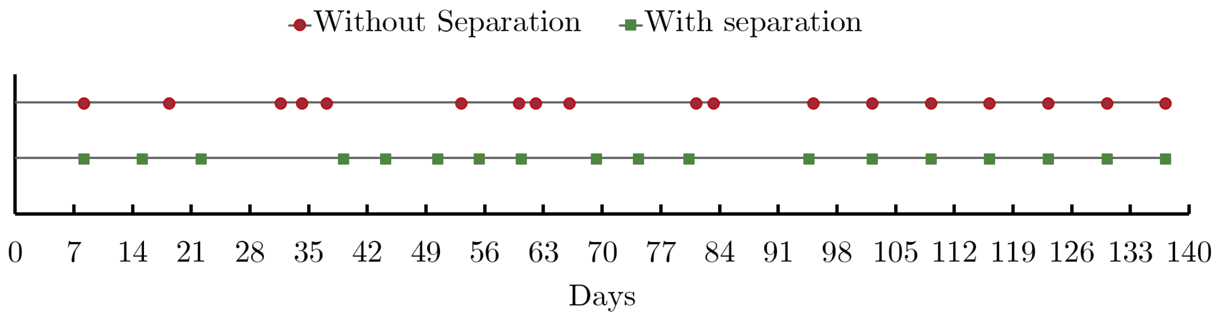

Figure 7.3: Starting times for voyages of a trade route.

Figure 7.3 shows that the spread of voyages is clearly better when the voyage separation requirement is applied. Without voyage separation there are several voyages that are clustered together on the time line, which in turn creates large gaps between voyages. This effect is undesired. Keep in mind that the time windows for each voyage are still in place, and will always ensure a minimum of spread of voyages throughout the planning horizon. Both with and without speed optimization have some intervals that are fairly long, as may be seen in Figure 7.3. The longest interval between two consecutive voyages for each approach are 16 and 17 days, respectively. 17 days are the longest possible interval for this given trade. Based om this, a possible addition to the given model could be to introduce an upper limit of separation. However, there is a narrow line between restricting the problem in order to make it more optimal, and making it infeasible.

7.6 Discussion of the Results

Up to this point, this chapter has mainly focused on the results achieved through the mathematical models. But there may be more that lies behind the results than what it may seem like from a comparison of results alone. This section aims to discuss some underlying considerations that could be important to understand the results. These considerations can be divided in two categories: managerial considerations and assumptions and limitations with the models.

7.6.1 Managerial Considerations

The results presented in this chapter show promising aspects of integrating speed optimization, however, it is not straightforward to apply these results in the real world. First of all, a schedule maker in a shipping company has numerous aspects to consider, when making ship schedules, not only the fleet deployment, even though it is the most important part. There are rules and regulations to abide, crew schedules, perhaps seasonality to account for and so on. Hence, the results in this thesis are based on a model for a part of the real challenges that a schedule maker faces. It is important to remember the overall picture. However, this thesis shows that optimization-based scheduling with integrated speed optimization may be a very useful tool.

One of the useful aspects of optimization-based scheduling is the fact that it achieves the best overall solution and consider all ships and routes at the same time. As shown in the results presented above, the computational complexity is vastly increasing as the instance size increases, especially when the time horizons are extended. The largest instances solved in this study has 18 ships. Out in the real world, this would be a fairly small fleet. The shipping market is difficult to enter, and there are several large players in the market. The computational study is based on data from the shipping company Saga Forrest Carriers. Since this data was collected, Saga Forrest Carriers has formed a joint shipping pool operation with Westfal-Larsen Shipping, administrated under the company Saga Welco (Holm, 2014). This means that the fleet composition that our data is based on does not exist anymore and the results achieved cannot directly be applied to a real-world case. However, the computational study is an exemplification of the general models, which in principle can be applied to any fleet size.

As shown in Section 7.4.2 the longest planning horizon that has been solved is 180 days, for instances with 10 ships only. Beyond this point, the instances became far too large and contain a huge amount of paths. The number of paths in Instance 16 and 17 give an indication of the enormous growth in number of paths. An obvious benefit of being able to use a longer planning horizon would have been to be able to make longer sailing schedules. Most shipping companies do not use that long schedules in their planning. What schedules with a longer planning horizon is actually useful for, is as basis in budgeting and forecasting for the company. Therefore, from a managerial point of view, longer planning horizons would be favourable.

In reality, the shipping industry and the global market are much more dynamic and nuanced, than what has been taken account for in this study. None of the models presented in this report consider uncertainty, even though there are numerous sources of uncertainty in the problem . Uncertainties that might be considered are related to, among others, weather conditions, fuel prices and the global economy with its regulations and sanctions. Of these, weather conditions and fuel prices can be considered short-term, whereas the global economy is a long-term uncertainty and less volatile. One should mostly consider the short-term uncertainties when reviewing the results. In a real-world situation, the models would be run several times within a planning horizon and data would be updated regularly and minimizing uncertainties. Updating the data does especially reduce the importance of long-term uncertainty in the model.

Another difference that need to be pointed out, is the difference in profit achieved with and without the heuristics shown in Section 7.3 and 7.4. If a shipping company would utilize optimization in their planning, they are interested in obtaining better fleet deployments than they can achieve through manual planning. An optimization-based solution would be far less interesting if it takes a long time to calculate the desired solution. In other words, a shipping company would prefer to have a good solution fast, rather than a marginally better solution in a and spending far more time on the schedule. This is exactly what implementing heuristics in the solution process does. As mentioned in Section 3.1, Höegh Autoliners applied an optimization-based model with heuristics, with good improvements compared to manual planning. However, it is not straightforward to implement heuristics into commercial software. In this computational study, the acceptance levels for each heuristic rule has been selected with care and based on extensive preliminary testing. In addition, this study has obtained results both with and without heuristics, and thus been able to compare and ensure the quality of the obtained results. The acceptance levels used in this study are customized for this specific data set, and not necessarily suitable for other cases. Thus, to achieve general, applicable heuristics as those presented in this study is not straightforward.

From a managerial point of view, the utilization of each ship is highly interesting. The profit is related to the utilization and net gross sailing of each ship. This thesis does not go into detail regarding utility specifically, but indirectly. The path reduction heuristics described in Section 5.2 removes paths that are undesirable, and thereby the heuristic rules indirectly remove paths with low utility. When these are put into combinations, all remaining paths results in reasonably well utilized ships. Hence, a side effect of applying heuristics is that a minimum of ship utility is achieved. However, there is not obtained any results to say something about actual utility of the ships.

The models in this thesis have an overall objective of maximizing profit from the shipping company's point of view. From a managerial perspective, not only is the profit important, but also the performance of the delivery is of high importance as well. If the shipping company does not deliver a service which satisfies the charterers, they will run out of business. An example where service performance is put up against profits, is when voyage separation is discussed in Section 7.5.1. Here, it is discussed that a lower profit with a better voyage separation should be regarded as a better solution, than one with higher profit and no guarantee of voyage separation. The trade-off between profit, which is concrete and easy to measure, and voyage separation, which is "soft" and hard to measure, is difficult. The most important part of the solution obtained by this thesis's models, is not the profit, but rather the actual fleet deployment. In the models presented above, voyage separation requirements are modelled as hard constraints, without flexibility. By relaxing the separation constraints in the problem, one gives more flexibility to the solver, and it may be able to find a fleet deployment that for various reasons may be preferable. A fleet deployment with minor breaches of the fairly evenly spread requirements, may yield good solutions. From a managerial standpoint, the interpretation of the fairly evenly spread term is challenging to put into strict constraints and threshold values.

7.6.2 Assumptions and Limitations

There are some assumptions and limitations in this study that should be mentioned. As described in Section 6, the fuel consumption function is assumed to be a second order polynomial function of speed. This polynomial has three ship specific coefficients which in reality varies from ship to ship. In this study, general estimates of reasonable value for an average ship in this kind of shipping are used for these coefficients. The same values are applied to all ships. As for the scaling factor used to calculate the reduced ballast fuel consumption, this is the same for all ships as well. In addition, the fuel consumption is dependent of other factors than speed as well. Such factors may be weather conditions, the amount of fouling on the hull, maintenance of the motor etc. Hence, it is impossible to have an exact fuel consumption function, and it will always be a source of inaccuracies, both in the linearization and in the function itself. In addition, each ship has a sailing speed interval, and can in theory sail with any speed in the interval. However, in practice the theoretical maximum speed can only be achieved in good weather conditions. Thus, in the execution of voyages, currents and weather conditions have an impact on service speed and hence fuel consumption. It is important to keep in mind that the intention of the analysis in this thesis is not to give exact data for a specific case, as mentioned earlier.

The data of the instances that the computational study is based on, are somewhat simplified. The data consider several parameters to be identical for all ships, although this is not necessarily true. All ships are assigned the same maximum and minimum speeds and all ships are compatible with all trades. As the ships are designed differently and have different engines they will in reality not have the same maximum and minimum speed. Some ships may be too big for a port, and thus all ships cannot be compatible with all trades. In addition, all ships have the same service time in ports for all voyage on a given trade route. When the ships have identical data, it will result in some symmetrical solutions. These assumptions and limitations are obvious sources of error in the calculations. However, as mentioned, the concepts of this thesis are universal, and all calculations are based on realistic data, which provides reasonable results and analyses.

Chapter 8

Concluding Remarks

In this thesis, an arc flow and a path flow formulation of the fleet deployment problem within the liner shipping segment have been presented. Solving the fleet deployment problem means to find the optimal deployment schedule for a fleet of ships. Both models integrate speed optimization and voyage separation requirements. The two models are implemented in the commercial solver Xpress IVE. The path flow model consists of a decomposition, into a master problem and ship specific subproblems. The subproblems are solved through a priori path generation of all feasible paths, using Matlab. The results presented show that the path flow model outperforms the arc flow model with regard to both solution time and profit.

To evaluate the speed optimization part of the problem, a posteriori speed optimization is performed on the fleet deployment solutions, both with and without speed optimization. Based on the results, the model was solved considerably faster when speed optimization was not applied. However, integrating speed optimization achieves higher profits. When the a posteriori speed optimization results in different profit, the underlying fleet deployment decisions have to be altered, thus the models with and without speed optimization obtains different schedules. There is a trade-off between solution time and profit that should be considered.

The number of paths in the path flow model increases with an exponential trend as the problem size are increased, which in turn increases the problem complexity and makes the problem harder to solve. To obtain good solutions faster, heuristic rules in the a priori path generation are implemented to reduce the total amount of paths. These heuristic rules are tested both individually and in combination for different acceptance levels, i.e. how strict the rules are enforced. Testing of these heuristics show that it is possible to reduce the number of paths drastically and still obtain solutions of high quality. Especially larger test instances, which are not solved to optimality without these heuristic rules, performs exceptionally well when the combinatorial heuristic rules are applied and obtains increased profits. For the largest instances up to 98 % of paths are removed, and improved results are still obtained. The largest instance that is solvable, and achieves a reasonable profit, with heuristics contains 18 ship, 9 trade routes, 79 voyages and a planning horizon of 150 days.

The voyage separation requirement is enforced to secure a fairly evenly spread in time between voyages. The effects of applying varying degrees of the voyage separation requirement are tested. Results show that applying the constraint causes a marginal reduction in profit with the benefit of a better separation between voyages. The combination of the voyage separation requirement and speed optimization is also tested. These tests show that the best separation is achieved when both speed optimization and voyage separation is applied, and the worst separation when only the speed optimization is applied. This shows that applying the voyage separation requirement should be of even higher importance when speed optimization is integrated with the problem.

The findings of the computational study prove that the path flow model with speed optimization and voyage separation should be considered a useful tool when solving the fleet deployment problem. As time moves on, the amount of available computational power will increase continuously, and so will the importance of utilizing optimization based scheduling within the shipping industry.

Chapter 9

Future Research

The field of maritime fleet deployment has been researched to some extent. However, there are still many interesting approaches to investigate further. The integrated problem with speed optimization and voyage separation requirements tend to become very large, and more sophisticated models and solution methods should be explored. A suggestion would be to use a branch-and-price algorithm, i.e. a dynamic column generation approach instead of generating the paths a priori. By using dynamic column generation, where only the most promising paths are generated and sent to the master problem. Such an approach may offer great advantages, especially for larger, real-world cases. Different approaches that solve the problem by using more advanced heuristics or other non-exact methods, such as metaheuristics, could also be interesting to look into. An example of a heuristic approach is the rolling horizon heuristic. In reality, the scheduling for a shipping company is a more or less continuous process, where the schedule is reoptimized and prolonged before the planning horizon is carried out. This resembles the way that the rolling horizon heuristic work, where the decisions for the entire planning horizon is divided into shorter time periods. First, the decisions for the first time period are taken, then the decisions for the next period is done based on the previous decisions, and so on.

The shipping industry is under constant uncertainty and influenced by several factors, as discussed in section 7.6. To implement uncertainty as a factor in the scheduling process may yield additional value of the solutions for the decision maker. A way to do this may be a robust optimization approach.

As for the models presented in this thesis, there are limitations and assumptions, as discussed in 7.6, that could be explored further in detail, in order to utilize the fleet capacity to the most and make the models even more realistic. The basis for the models is to serve a given number of voyages on a given set of trades. The models do not distinguish between ships when it comes to load capacities or the requested demand per voyage. Two different approaches to accommodate these limitations are suggested. First, a model that implements that each voyage has a given demand, where demand covering restrictions may be added. By including this, the model ensures that a suitable sized ship serves each voyage, and hence less overcapacity. Second, if demands are introduced, stochastic analysis and forecasting could make the basis for the decision making stronger, and more adaptable to a rapidly changing market.

Bibliography

Alvarez, J. F., Longva, T., and Engebrethsen, E. S. (2010). A methodology to assess vessel berthing and speed optimization policies. *Maritime economics & logistics*, 12(4):327–346.

Andersson, H., Fagerholt, K., and Hobbesland, K. (2015). Integrated maritime fleet deployment and speed optimization: Case study from roro shipping. *Computers & Operations Research*, 55:233–240.

Asariotis, R., Assaf, M., Benamara, H., Fugazza, M., Hoffmann, J., Premti, A., Rodriguez, L., Ugaz, P., Weller, M., and Youssef, F. (2017). Review of maritime transport, 2017. Technical report. `http://unctad.org/en/PublicationsLibrary/rmt2017_en.pdf` [Accessed: 20.05.2018.].

Asariotis, R., Benamara, H., Hoffmann, J., Premti, A., Valentine, V., and Youssef, F. (2016). Review of maritime transport, 2016. Technical report. `http://unctad.org/en/PublicationsLibrary/rmt2016_en.pdf` [Accessed: 20.05.2018.].

Bakkehaug, R., Rakke, J. G., Fagerholt, K., and Laporte, G. (2016). An adaptive large neighborhood search heuristic for fleet deployment problems with voyage separation requirements. *Transportation Research Part C: Emerging Technologies*, 70:129–141.

Bélanger, N., Desaulniers, G., Soumis, F., Desrosiers, J., and Lavigne, J. (2006). Weekly airline fleet assignment with homogeneity. *Transportation Research Part B*, 40(4):306–318.

Borander, V., Lunde, S., and Straume, A. (2017). A study of maritime fleet deployment with integrated speed optimization. Project assignment at NTNU.

Borsani, V., Matta, A., Beschi, G., and Sommaruga, F. (2006). A home care scheduling model for human resources. In *Service systems and service management, 2006 international conference on*, volume 1, pages 449–454. IEEE.

Cames, M., Graichen, J., Siemons, A., and Cook, V. (2015). Emission reduction targets for international aviation and shipping. *Study for the ENVI Committee. Directorate General for Internal Policies Policy Department a: Economic and Scientific Policy. European Parliament.*

Christiansen, M., Fagerholt, K., Nygreen, B., and Ronen, D. (2007). Maritime transportation. *Handbooks in operations research and management science*, 14:189–284.

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

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

Dohn, A., Rasmussen, M. S., and Larsen, J. (2011). The vehicle routing problem with time windows and temporal dependencies. *Networks*, 58(4):273–289.

Dong, B., Bektas, T., Chandra, S., Christiansen, M., and Fagerholt, K. (2017). A new formulation for the combined maritime fleet deployment and inventory management problem. *Lecture Notes in Computer Science - Computational Logistics (Eds: Bektas, Coniglio, Martinez-Sykora and Voss)*, 10572:321–335.

Fagerholt, K., Gausel, N. T., Rakke, J. G., and Psaraftis, H. N. (2015). Maritime routing and speed optimization with emission control areas. *Transportation Research Part C: Emerging Technologies*, 52:57–73.

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

Fagerholt, K., Laporte, G., and Norstad, I. (2010). Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3):523–529.

Fischer, A., Nokhart, H., Olsen, H., Fagerholt, K., Rakke, J. G., and Stålhane, M. (2016). Robust planning and disruption management in roll-on roll-off liner shipping. *Transportation Research Part E: Logistics and Transportation Review*, 91:51–67.

Fisher, M. L. and Rosenwein, M. B. (1989). An interactive optimization system for bulk-cargo ship scheduling. *Naval Research Logistics (NRL)*, 36(1):27–42.

Gelareh, S. and Meng, Q. (2010). 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.

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

Holm, M. (2014). Twestfal-larsen shipping går sammen med saga. `https://sysla.no/maritim/westfal-larsen-shipping-gar-sammen-med-saga/` [Accessed: 15.12.2017.].

Hvattum, L. M., Norstad, I., Fagerholt, K., and Laporte, G. (2013). Analysis of an exact algorithm for the vessel speed optimization problem. *Networks*, 62(2):132–135.

IMO (n.d.). Imo overview. `https://business.un.org/en/entities/13` [Accessed: 10.12.2017.].

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

Nicholson, T. and Pullen, R. (1971). Dynamic programming applied to ship fleet management. *Operational Research Quarterly*, pages 211–220.

Norstad, I., Fagerholt, K., Hvattum, L. M., Arnulf, H. S., and Bjørkli, A. (2015). Maritime fleet deployment with voyage separation requirements. *Flexible Services and Manufacturing Journal*, 27(2-3):180–199.

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

Psaraftis, H. N. and Kontovas, C. A. (2014). Ship speed optimization: Concepts, models and combined speed-routing scenarios. *Transportation Research Part C: Emerging Technologies*, 44:52–69.

Redjem, R., Kharraja, S., Xie, X., and Marcon, E. (2012). Routing and scheduling of caregivers in home health care with synchronized visits. In *9th International Conference on Modeling, Optimization & Simlation*.

Reinhardt, L. B., Plum, C. E., Pisinger, D., Sigurd, M. M., and Vial, G. T. (2016). The liner shipping berth scheduling problem with transit times. *Transportation Research Part E: Logistics and Transportation Review*, 86:116–128.

Ronen, D. (1982). The effect of oil price on the optimal speed of ships. *Journal of the Operational Research Society*, 33(11):1035–1040.

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

Ship & Bunker (n.d.). Global 20 ports average bunker price. `https://shipandbunker.com/prices/av/global/av-g20-global-20-ports-average` [Accessed: 17.04.2018].

Sigurd, M. M., Ulstein, N. L., Nygreen, B., and Ryan, D. M. (2005). Ship scheduling with recurring visits and visit separation requirements. *Column generation*, pages 225–245.

Smith, T., Jalkanen, J., Anderson, B., Corbett, J., Faber, J., Hanayama, S., O'Keeffe, E., Parker, S., Johansson, L., Aldous, L., et al. (2014). Third imo greenhouse gas study 2014. *International Maritime Organization*, 327.

Transport & Environment (n.d.). Maritime fleet deployment with voyage separation requirements. `https://www.transportenvironment.org/what-we-do/shipping/shipping-and-climate-change` [Accessed: 14.12.2017.].

Vilhelmsen, C., Lusby, R. M., and Larsen, J. (2017). Tramp ship routing and scheduling with voyage separation requirements. *OR Spectrum*, 39(4):913–943.

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

Wang, S., Wang, T., and Meng, Q. (2011). A note on liner ship fleet deployment. *Flexible Services and Manufacturing Journal*, 23(4):422–430.

Wen, M., Ropke, S., Petersen, H. L., Larsen, R., and Madsen, O. B. (2016). Full-shipload tramp ship routing and scheduling with variable speeds. *Computers & Operations Research*, 70:1–8.

Appendix A

Arc Flow Formulation

A.1 Notation

Sets

$\mathcal{V}$	Set of ships treated individually.
$\mathcal{R}$	Set of trade routes.
$\mathcal{R}_v$	Set of trade routes that ship v can sail.
$\mathcal{N}$	Set of four types of nodes: Origin nodes, destination nodes, voyage nodes and maintenance nodes.
$\mathcal{A}$	Set of arcs.
$\mathcal{A}_v$	Set of arcs that ship v can service.
$\mathcal{N}^C$	Set of contracted (compulsory) voyage nodes.
$\mathcal{N}^O$	Set of spot (optional) voyage nodes.
$\mathcal{N}_v^M$	Set of nodes that correspond to required ship yard maintenance for ship v.
$\mathcal{N}_v$	Set of all nodes in $\mathcal{N}$ that correspond to voyages that ship v can service.
$\mathcal{I}_r$	Set of voyages on trade route r.
$\mathcal{S}$	Set of speed alternatives.

Parameters

T^B_{vriqjs}	The time it takes ship v to sail the arc $((r,i),(q,j))$ with speed alternative s.
C^B_{vriqjs}	The cost corresponding to sail the arc $((r,i),(q,j))$ for ship v with speed alterative s.
$T^B_{vo(v)ris}$	The time it takes ship v to sail from its origin $o(v)$ to the start of voyage(r,i) with speed alternative s.
$C^B_{vo(v)ris}$	The cost corresponding to sail with speed s from its origin $o(v)$ to the start of voyage(r,i) for ship v.
T_{vris}	The time it takes ship v to sail the voyage (r,i) with speed alternative s.
C_{vris}	The cost corresponding to sail the voyage (r,i) with speed s for ship v.
R_{ri}	The revenue, freight income minus port costs for a given voyage (r,i)
C^S_{ri}	The cost of chartering a spot ship to service voyage (r,i) on trade r.
E_{ri}	The earliest time for starting voyage i on trade r.
$E_{o(v)}$	The earliest time ship v can leave its origin $o(v)$.
L_{ri}	The latest time for starting voyage i on trade r.
B_r	The minimum accepted time between two consecutive voyages on trade r.

Variables

x_{vriqj}	1 if ship v travels directly from node (r,i) to node (q,j), 0 otherwise
$x_{vo(v)ri}$	1 if ship v travels from its origin node $o(v)$ to node (r,i), 0 otherwise
$x_{vrid(v)}$	1 if node (r,i) is the last node ship v visits before $d(v)$,
$x_{vo(v)d(v)}$	1 if ship v does not service any voyages at all, 0 otherwise 0 otherwise
w^B_{vriqjs}	The weight of sailing with speed alternative s for ship v

on voyage (r,i).

from node (r,i) to node (q,j)

w_{vris} The weight of sailing with speed alternative s for ship v on voyage (r,i)

$w^B_{vo(v)ris}$ The weight of sailing with speed alternative s for ship v from origin node $o(v)$ to node (r,i),

u^S_{ri} 1 if voyage i on trade r is serviced by a chartered spot ship, 0 otherwise

t_{ri} The time for start of voyage i of trade r.

A.2 The model

$$\begin{aligned}\max &\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{s\in\mathcal{S}}(R_{ri}-C_{vris})w_{vris}\\ &-\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{s\in\mathcal{S}}C^B_{vo(v)ris}w_{vo(v)ris}\\ &-\sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}\sum_{s\in\mathcal{S}}C^B_{vriqjs}w^B_{vriqjs}+\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{I}_r}(R_{ri}-C^S_{ri})u^S_{ri}\end{aligned} \tag{A.1}$$

s.t.

$$\sum_{v\in\mathcal{V}_r}\left[\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}x_{vriqj}+x_{vrid(v)}\right]+u^S_{ri}=1,\quad (r,i)\in\mathcal{N}^C \tag{A.2}$$

$$\sum_{v\in\mathcal{V}_r}\left[\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}x_{vriqj}+x_{vrid(v)}\right]\leq 1,\quad (r,i)\in\mathcal{N}^O \tag{A.3}$$

$$\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}x_{vriqj}+x_{vrid(v)}=1,\quad v\in\mathcal{V},(r,i)\in\mathcal{N}^M_v \tag{A.4}$$

$$x_{vo(v)d(v)}+\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}x_{vo(v)ri}=1,\quad v\in\mathcal{V} \tag{A.5}$$

$$\begin{aligned}&x_{vrid(v)}+\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q}x_{vriqj}-\sum_{q\in\mathcal{R}_v}\sum_{j\in I_q}x_{vqjri}-x_{o(v)ri}=0,\\ &\quad v\in\mathcal{V},r\in\mathcal{R}_v,i\in\mathcal{I}_r\end{aligned} \tag{A.6}$$

$$x_{vo(v)d(v)} + \sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r} x_{rid(v)} = 1, \quad v \in \mathcal{V} \tag{A.7}$$

$$x_{vo(v)ri} - \sum_{s\in\mathcal{S}} w_{o(v)ris} = 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{A.8}$$

$$x_{vriqj} - \sum_{s\in\mathcal{S}} w_{vriqjs} = 0, \quad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \tag{A.9}$$

$$x_{vrid(v)} + \sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_q} x_{vriqj} - \sum_{s\in\mathcal{S}} w_{vris} = 0, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{A.10}$$

$$\begin{aligned} &E_{o(v)} + \sum_{s\in\mathcal{S}} T^B_{vo(v)ris} w_{vo(v)ris} - t_{ri} - E_{o(v)}(1 - x_{vo(v)ri}) \leq 0, \\ &v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \end{aligned} \tag{A.11}$$

$$\begin{aligned} &t_{ri} + \sum_{s\in\mathcal{S}} (T_{vris} w_{vris} + T^B_{vriqjs}) w_{vriqjs} - t_{qj} - L_{ri}(1 - x_{vriqj}) \leq 0, \\ &v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \end{aligned} \tag{A.12}$$

$$E_{ri} \leq t_{ri} \leq L_{ri}, \quad r \in \mathcal{R}, i \in \mathcal{I}_r \tag{A.13}$$

$$t_{r,i+1} - t_{ri} \geq B_r, \quad r \in \mathcal{R}, i \in \mathcal{I}_r \backslash \{n_r\} \tag{A.14}$$

$$x_{vo(v)d(v)} \in \{0,1\}, \quad v \in \mathcal{V} \tag{A.15}$$

$$x_{vo(v)ri} \in \{0,1\}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{A.16}$$

$$x_{vrid(v)} \in \{0,1\}, \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{A.17}$$

$$x_{vriqj} \in \{0,1\}, \quad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v \tag{A.18}$$

$$w_{vo(v)ris} \in [0,1], \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, s \in \mathcal{S} \tag{A.19}$$

$$w_{vris} \in [0,1], \quad v \in \mathcal{V}, r \in \mathcal{R}_v, i \in \mathcal{I}_r, s \in \mathcal{S} \tag{A.20}$$

$$w_{vriqjs} \in [0,1], \quad v \in \mathcal{V}, ((r,i),(q,j)) \in \mathcal{A}_v, s \in \mathcal{S} \tag{A.21}$$

$$t_{ri} > 0, \quad r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{A.22}$$

$$u^S_{ri}, \in \{0,1\} \quad r \in \mathcal{R}_v, i \in \mathcal{I}_r \tag{A.23}$$

Appendix B

Path Flow Formulation

B.1 Notation

Sets

$\mathcal{V}$	Set of ships treated individually.
$\mathcal{R}$	Set of trade routes.
$\mathcal{R}_v$	Set of trade routes that ship v can sail.
$\mathcal{N}$	Set of four types of nodes: Origin nodes, destination nodes, voyage nodes and maintenance nodes.
$\mathcal{A}$	Set of arcs.
$\mathcal{A}_v$	Set of arcs that ship v can service.
$\mathcal{N}^C$	Set of contracted (compulsory) voyage nodes.
$\mathcal{N}^O$	Set of spot (optional) voyage nodes.
$\mathcal{N}_v^M$	Set of nodes that correspond to required ship yard maintenance for ship v.
$\mathcal{N}_v$	Set of all nodes in $\mathcal{N}$ that correspond to voyages that ship v can service.
$\mathcal{I}_r$	Set of voyages on trade route r.
$\mathcal{S}$	Set of speed alternatives.
$\mathcal{P}_v$	Set of all feasible paths for ship v.

$\mathcal{P}_{vri}$ Set of all paths where ship v sails voyage i on route r.

$\mathcal{P}_{vo(v)ri}$ Set of all paths where voyage i on route r is the first voyage that ship v performs from its origin.

$\mathcal{P}_{vriqj}$ Set of all paths where voyage j on trade route q directly follows voyage i on route r for ship v.

Parameters

T^B_{vriqjs} The time it takes ship v to sail the arc $((r,i),(q,j))$ with speed alternative s.

C^B_{vriqjs} The cost corresponding to sail the arc $((r,i),(q,j))$ for ship v with speed alterative s.

$T^B_{vo(v)ris}$ The time it takes ship v to sail from its origin $o(v)$ to the start of voyage(r,i) with speed alternative s.

$C^B_{vo(v)ris}$ The cost corresponding to sail with speed s from its origin $o(v)$ to the start of voyage(r,i) for ship v.

T_{vris} The time it takes ship v to sail the voyage (r,i) with speed alternative s.

C_{vris} The cost corresponding to sail the voyage (r,i) with speed s for ship v.

R_{ri} The revenue, freight income minus port costs for a given voyage (r,i)

C^S_{ri} The cost of chartering a spot ship to service voyage (r,i) on trade r.

E_{ri} The earliest time for starting voyage i on trade r.

E_{vpri} The earliest starting time for ship v on voyage i on trade route r for a given path p.

$E_{o(v)}$ The earliest time ship v can leave its origin $o(v)$.

L_{ri} The latest time for starting voyage i on trade r.

B_r The minimum accepted time between two consecutive voyages on trade r.

Variables

z_{vp}	1 if ship v sail path p, and 0 otherwise.
t_{vri}	The start time of voyage i on route r for ship v.
t^S_{ri}	The start time of a spot ship for sailing voyage i on route r.
w^B_{vriqjs}	The weight of sailing with speed alternative s for ship v on voyage (r,i). from node (r,i) to node (q,j)
w_{vris}	The weight of sailing with speed alternative s for ship v on voyage (r,i)
$w^B_{vo(v)ris}$	The weight of sailing with speed alternative s for ship v from origin node $o(v)$ to node (r,i),
u^S_{ri}	1 if voyage i on trade r is serviced by a chartered spot ship, 0 otherwise

B.2 The model

$$\max \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}_v}\sum_{i\in\mathcal{I}_r}\left[\sum_{s\in\mathcal{S}}(R_{ri}-C_{vris})w_{vris}-\sum_{q\in\mathcal{R}_v}\sum_{j\in\mathcal{I}_r}\sum_{s\in\mathcal{S}}C^B_{vriqjs}w^B_{vriqjs}\right.$$
$$\left.-\sum_{s\in\mathcal{S}}C^B_{vo(v)ris}w^B_{vo(v)ris}\right]+\sum_{r\in\mathcal{R}}\sum_{i\in\mathcal{I}_r}(R_{ri}-C^S_{ri})u^S_{ri} \quad \text{(B.1)}$$

s.t.

$$\sum_{v\in\mathcal{V}_r}\sum_{p\in\mathcal{P}_{vri}} z_{vp}+u^S_{ri}=1, \quad (r,i)\in\mathcal{N}^C \quad \text{(B.2)}$$

$$\sum_{v\in\mathcal{V}_r}\sum_{p\in\mathcal{P}_{vri}} z_{vp}\leq 1, \quad (r,i)\in\mathcal{N}^O \quad \text{(B.3)}$$

$$\sum_{p\in\mathcal{P}_v} z_{vp}=1, \quad v\in\mathcal{V} \quad \text{(B.4)}$$

$$\sum_{s\in\mathcal{S}} w_{vris}=\sum_{p\in\mathcal{P}_{vri}} z_{vp}, \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r \quad \text{(B.5)}$$

$$\sum_{s\in\mathcal{S}} w^B_{vo(v)ris}=\sum_{p\in\mathcal{P}_{vo(v)ri}} z_{vp}, \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r \quad \text{(B.6)}$$

$$\sum_{s\in\mathcal{S}} w^B_{vriqjs} = \sum_{p\in\mathcal{P}_{vriqj}} z_{vp}, \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r, q\in\mathcal{R}_v, j\in\mathcal{I}_q \tag{B.7}$$

$$\sum_{p\in\mathcal{P}_{vri}} E_{vpri} z_{vp} \leq t_{vri} \leq \sum_{p\in\mathcal{P}_{vri}} L_{ri} z_{vp}, \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r \tag{B.8}$$

$$E_{ri} u^S_{ri} \leq t^S_{ri} \leq L_{ri} u^S_{ri}, \quad r\in\mathcal{R}, i\in\mathcal{I}_r \tag{B.9}$$

$$\sum_{s\in\mathcal{S}} \big(T^B_{vo(v)ris} + E_{o(v)}\big) w^B_{vo(v)ris} \leq t_{vri}, \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r \tag{B.10}$$

$$\begin{aligned} &t_{vri} + \sum_{s\in\mathcal{S}} \big(T_{vris} w_{vris} + T^B_{vriqjs} w^B_{vriqjs} + (L_{ri} + T_{vri,1}) w^B_{vriqjs}\big) \\ &- L_{ri} - T_{vri,1} - t_{vqj} \leq 0, \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r, q\in\mathcal{R}_v, j\in\mathcal{I}_q \end{aligned} \tag{B.11}$$

$$B_r + \sum_{v\in\mathcal{V}} t_{vri} + t^S_{ri} - \sum_{v\in\mathcal{V}} t_{vr,i+1} - t^S_{r,i+1} \leq 0, \quad r\in\mathcal{R}, i\in\mathcal{I}_r\backslash\{n_r\} \tag{B.12}$$

$$z_{vp} \in \{0,1\}, \quad v\in\mathcal{V}, p\in\mathcal{P}_v \tag{B.13}$$

$$u^S_{ri} \in \{0,1\}, \quad r\in\mathcal{R}, i\in\mathcal{I}_r \tag{B.14}$$

$$w_{vris} \in [0,1], \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r, s\in\mathcal{S} \tag{B.15}$$

$$w^B_{vriqjs} \in [0,1], \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r, q\in\mathcal{R}_v, j\in\mathcal{I}_q, s\in\mathcal{S} \tag{B.16}$$

$$w^B_{vo(v)ris} \in [0,1], \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r, s\in\mathcal{S} \tag{B.17}$$

$$t^S_{ri} > 0, \quad r\in\mathcal{R}_v, i\in\mathcal{I}_r \tag{B.18}$$

$$t_{vri} > 0, \quad v\in\mathcal{V}, r\in\mathcal{R}_v, i\in\mathcal{I}_r \tag{B.19}$$

Appendix C

Tables

Heuristic Combinations

Table C.1: Heuristic combinations with acceptance levels.

Comb.	Max. Length Ballast Sailing	Max. Consecutive Waiting Days	Max. % Ballast Sailing
A	10,500	30	45 %
B	10,500	30	40 %
C	10,500	30	35 %
D	10,500	30	30 %
E	10,500	20	45 %
F	10,500	20	40 %
G	10,500	20	35 %
H	10,500	20	30 %
I	10,500	10	45 %
J	10,500	10	40 %
L	10,500	10	35 %
L	10,500	10	30 %
M	10,000	30	45 %
N	10,000	30	40 %
O	10,000	30	35 %
P	10,000	30	30 %
Q	10,000	20	45 %
R	10,000	20	40 %
S	10,000	20	35 %
T	10,000	20	30 %
U	10,000	10	45 %
V	10,000	10	40 %
W	10,000	10	35 %
X	10,000	10	30 %

Results of Heuristic Combinations

Table C.2: Results of all heuristic combinations.

	Instance 6			Instance 9			Instance 12			Average	
Comb.	Paths	Profit	% Impr	Paths	Profit	% Impr.	Paths	Profit	% Impr.	Profit	% Impr.
None	3,277	28,926'	-	69,776	42,227'	-	138,392	51,161'	-	40,771'	-
A	1,213	28,327'	-1.63%	24,737	42,655'	1.01%	52,925	51,325'	0.32%	40,769'	-0.10%
B	1,006	28,327'	-1.63%	16,463	42,653'	1.01%	37,888	51,214'	0.10%	40,731'	-0.17%
C	746	27,946'	-2.95%	10,424	43,349'	2.66%	25,815	51,710'	1.07%	41,001'	0.26%
D	535	27,012'	-6.19%	5,074	42,845'	1.46%	15,354	52,012'	1.66%	40,623'	-1.02%
E	970	28,327'	-1.63%	19,603	42,716'	1.16%	42,338	51,841'	1.33%	40,961'	0.29%
F	798	28,327'	-1.63%	12,721	43,180'	2.26%	29,782	51,218'	0.11%	40,908'	0.25%
G	596	27,946'	-2.95%	8,127	43,075'	2.01%	29,782	52,098'	1.83%	41,040'	0.30%
H	430	26,670'	-7.38%	3,944	42,700'	1.12%	11,972	52,148'	1.93%	40,506'	-1.44%
I	600	28,240'	-1.93%	11,884	43,121'	2.12%	26,420	51,493'	0.65%	40,951'	0.28%
J	487	28,240'	-1.93%	7,542	43,208'	2.32%	18,294	52,305'	2.24%	41,251'	0.88%
L	371	27,304'	-5.18%	4,765	43,567'	3.17%	12,271	51,978'	1.60%	40,950'	-0.14%
L	270	26,659'	-7.42%	2,408	42,713'	1.15%	7,115	52,632'	2.87%	40,668'	-1.13%
M	1,000	27,920'	-3.04%	22,065	42,830'	1.43%	47,471	50,490'	-1.31%	40,413'	-0.97%
N	866	27,920'	-3.04%	15,335	42,772'	1.29%	35,200	51,588'	0.83%	40,760'	-0.30%
O	713	27,641'	-4.01%	10,013	42,905'	1.61%	24,656	51,918'	1.48%	40,821'	-0.31%
P	533	27,012'	-6.19%	5,003	42,888'	1.56%	15,091	52,321'	2.27%	40,740'	-0.79%
Q	801	26,937'	-6.45%	17,297	41,523'	-1.67%	37,601	51,684'	1.02%	40,048'	-2.37%
R	688	26,937'	-6.45%	11,768	42,168'	-0.14%	27,473	51,919'	1.48%	40,341'	-1.70%
S	565	26,670'	-7.38%	7,760	42,194'	-0.08%	19,433	51,981'	1.60%	40,282'	-1.95%
T	428	26,670'	-7.38%	3,876	42,144'	-0.20%	11,738	52,921'	3.44%	40,578'	-1.38%
U	510	26,914'	-6.53%	10,431	41,859'	-0.87%	23,210	52,083'	1.80%	40,285'	-1.87%
V	435	26,914'	-6.53%	6,979	42,361'	0.32%	16,759	51,965'	1.57%	40,413'	-1.55%
W	357	26,659'	-7.42%	4,563	42,302'	0.18%	11,639	52,103'	1.84%	40,354'	-1.80%
X	268	26,659'	-7.42%	2,365	42,113'	-0.27%	6,956	52,740'	3.09%	40,504'	-1.53%