Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

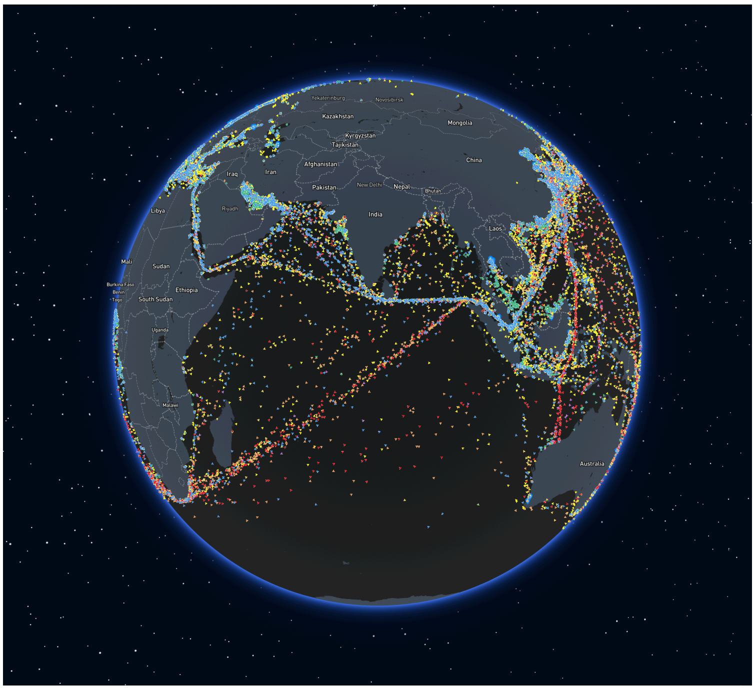
Simen Omholt-Jensen

Tramp Ship Routing and Scheduling in the Dry Bulk Industry

including Bunker Optimization and Fleet Repositioning

Master's thesis in Industrial Economics & Technology Management
Supervisor: Kjetil Fagerholt
Co-supervisor: Frank Meisel

June 2023



NTNU
Norwegian University of
Science and Technology

Simen Omholt-Jensen

Tramp Ship Routing and Scheduling in the Dry Bulk Industry

including Bunker Optimization and Fleet Repositioning

Master's thesis in Industrial Economics & Technology Management
Supervisor: Kjetil Fagerholt
Co-supervisor: Frank Meisel
June 2023

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

NTNU | Norwegian University of Science and Technology

Department of Industrial Economics & Technology Management

TIØ4905 - Managerial Economics and Operations Research, Master Thesis

Tramp Ship Routing and Scheduling in the Dry Bulk Industry

including Bunker Optimization and Fleet Repositioning

Author:
Simen Omholt-Jensen

Supervisors:
Kjetil Fagerholt
Frank Meisel

June, 2023

Preface

This master's thesis is written in the spring of 2023. It concludes my Master of Science in Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU), Department of Industrial Economics and Technology Management. The thesis builds on the specialization project written in the fall of 2022 within the subject TIØ4500 - Managerial Economics and Operations Research. This master thesis concern Tramp Ship Routing and Scheduling Problems with Flexible Cargo Quantities and Bunker Optimization and originated through discussion with industry partners Western Bulk and Maritime Optima.

I express my deepest gratitude to my supervisors, Prof. Dr. Kjetil Fagerholt (NTNU) and Prof. Dr. Frank Meisel (Christian-Albrechts-Universität zu Kiel). Their perpetual support and guidance have been essential for the success of this master thesis. In addition, I am highly grateful to the industry partner Western Bulk, in particular Egil Husby, who has provided real-life data and answers to questions regarding their operation. I also thank Maritime Optima and Jean Niklas L'orange for routing data. Finally, a special mention goes to Kjetil Fagerholt, who accepted to take on a student-defined research project as he rarely does.

Simen Omholt-Jensen

Trondheim, June 2022

Abstract

The maritime transport industry plays a crucial role in the global economy, with over 11 billion metric tonnes of goods traded by sea in 2019. Within this complex and diverse industry, dry bulk operators like Western Bulk exploit regional arbitrage opportunities when offering cargo transport services. This research focuses on a specific challenge faced by dry bulk operators, such as Western Bulk, known as the Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO). This problem aims to maximize the overall profit of a fleet of vessels by selecting the most profitable cargoes from a pool of available options. The formulation of this problem incorporates flexible cargo limits and integrated bunker optimization. Additionally, the study addresses the regional allocation of vessels and fleet repositioning. Consequently, a solution to this problem must determine the appropriate cargo quantities to transport, make routing and scheduling decisions regarding cargo and bunker ports, select the optimal bunker quantity to procure from each port and pinpoint the regions in which vessels should be located at the end of the planning period.

This thesis models the studied TSRSPBO as a two-stage stochastic optimization problem, where routing and bunkering decisions are solved in the first-stage problem, and the recourse cost of fleet repositioning is considered in the second-stage. An arc flow and a path flow model of the TSRSPBO are provided. To solve realistic test instances, the path flow solution method leverages an a priori column generation approach, generating and optimizing the set of all feasible routes. Results indicate that the path flow solution approach can optimally solve test instances of up to 30 cargoes, ten vessels, and ten bunker ports within one hour.

Western Bulk operates a long-term fleet of 30 vessels. To solve test instances of this size, an iterative matheuristic solution approach leveraging an Adaptive Large Neighborhood Search (ALNS) framework is proposed. By utilizing the ALNS framework to generate columns, the iterative matheuristic solves the path flow model at regular intervals to obtain high-quality solutions. Results show that the iterative matheuristic finds near-optimal solutions for small- and medium-sized test instances. Furthermore, the matheuristic successfully solves test instances of 120 cargoes, 30 vessels, and ten bunker ports in under one hour.

Finally, this thesis conducts an economic analysis of Western Bulk's operational environment. The added value of fleet repositioning considerations is shown to be significant. Additionally, this thesis quantifies the impact of procuring discounted bunker prices at specific ports, offering valuable decisional support to Western Bulk's chartering managers and bunker procurement department.

Sammendrag

Den maritime shippingindustrien er en bærebjelke for den globale økonomien. I 2019, ble mer enn 11 milliarder tonn med varer transportert av skip. I denne komplekse og mangfoldige industrien finnes tørrbulk operatører som Western Bulk. Som tilbydere av transporttjenester utnytter de regionale arbitrasjemuligheter for å maksimere profitt generert av en flåte med skip. Denne masterstudien utforsker et problem som er relevant for operatører som Western Bulk, definert som et Tramp Ship Routing and Scheduling Problem med Bunker Optimization (TSRSPBO). Målet til problemet er å maksimere profitten til en flåte med skip ved å velge ut hvilke laster man bør transportere. Problemformuleringen inkluderer fleksible lastekvoter, integrert bunkeroptimiering og håndterer i tillegg hvilke regioner skip bør ende opp i ved slutten av planleggingshorisonten ved hjelp av reposisjonering av flåten. Derfor må en løsning til et slikt problem definere mengden last som skal fraktes, bestemme hvilke laste- og bunkerhavner som skal besøkes, og allokere skip til de mest optimale regioner.

Denne masterstudien formulerer et TSRSPBO som en to-stegs stokastisk optimeringsproblem, der rute- og bunkerbeslutninger løses i første steg. I andre steg håndteres kostnaden av å ende opp i feil regioner. Både en modell basert på buer, og en modell basert på veier, blir definert. For å løse realistiske probleminstanser benytter modellen basert på veier seg av a priori kolonnegenerering. Alle mulige ruter blir først generert og så optimert. Deretter plukker modellen basert på veier ut de rutene som maksimerer flåteprofitten. Resultatene fra de gjennomførte eksperimentene viser at denne løsningsmetoden optimalt løser testinstanser med opp til 30 laster, ti skip og ti bunker havner innen en time.

Western Bulk opererer en langtidsflåte med 30 skip. For å løse så store testinstanser foreslår denne masterstudien en iterativ matheuristikk som utnytter et Adaptive Large Neighborhood Search (ALNS) for å generere kolonner. Den iterative matheuristikken løser modellen basert på veier ved faste mellomrom basert på kolonnene generert av ALNS. Resulter viser at matheuristikken finner nær optimale løsninger for små og mellomstore testinstanser. Matheuristikken finner også løsninger til store testinstanser med 120 laster, 30 skip og ti bunkerhavner innen en time.

Avslutningsvis gjennomfører denne masterstudien en økonomisk analyse av Western Bulk sitt operasjonelle miljø. Merverdien av å håndtere flåtereposisjonering viser seg å være betydelig. I tillegg kvantifiserer denne studien effekten av bunkerinnkjøp til priser med ulike avslag. Disse innsiktene viser at denne masterstudien tilbyr verdifull beslutningsstøtte til både Western Bulks befraktningssjefer og til deres avdelinger som bestemmer innkjøp av bunker.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Terms and Acronyms

Glossary

DWT Deadweight tonnage. A measure of how much weight a vessel can carry. It is the sum of the weights of the of cargo, fuel, fresh water, ballast water, provisions, passengers, and crew

Reefer ship Refrigerated cargo vessel typically used to transport perishable cargo

Acronyms

ALNS Adaptive Large Neighborhood Search
ALNS-VCP Adaptive Large Neighborhood Search for the Vessel Combination Problem

BDI Baltic Dry Index
BP Branch-and-Price

CoA Contract of Affreightment

DFS Depth-First-Search
DWT Deadweight Tonnage

FFA Forward Freight Agreement

IFO380 Intermediate Fuel Oil with Maximum Viscosity of 380 Centistokes

JIT Just-In-Time

LNS Large Neighborhood Search
LP Linear Programming
LS Local Search

MILP Mixed-Integer Linear Problem
MoLOO More or Less in Owner's Option

PSD Parcel Size Distribution

Ro-Ro Roll-on, Roll-off

SPPRC Shortest Path Problem with Resource Constraints
SRS Ship Routing and Scheduling

TC Time Charter
TSRSP Tramp Ship Routing and Scheduling Problem
TSRSPBO Tramp Ship Routing and Scheduling Problem with Bunker Optimization

UHGS Unified Hybrid Genetic Search

VCP Vessel Combination Problem
VLSFO Very Low Sulfur Fuel Oil
VRP Vehicle Routing Problem
VRPTW Vehicle Routing Problem with Time Windows
VSP Vessel Scheduling Problem

Chapter 1

Introduction

Maritime transportation concerns transporting cargo around the globe using the world's seaways. It serves as a fundamental pillar of global trade. Maritime transportation dominates other modes of transportation in terms of the total volume of goods transported. In 2019, more than 80% of the internationally traded volume was carried by ships (UNCTAD, 2022). Factors like absolute advantages, comparative advantages, technological advancements, and economies of scale contribute to this scale. The key participants in the maritime transportation industry include cargo owners, commodity traders, shipowners, and bulk operators. Cargo owners possess production facilities, while commodity traders take advantage of regional price discrepancies by buying and selling goods. Shipowners treat vessels as assets, speculating on market trends. Bulk operators, on the other hand, buy and sell cargo transportation services by chartering vessels and securing cargo. They act as intermediaries between shipowners and cargo owners and typically do not own the ships they manage. Instead, they operate within the margins between operational costs and revenue generated from transportation. This line of business involves significant risks. However, a deep understanding of trade patterns and skills in securing transport contracts and managing cargo can lead to financial success. The maritime transportation industry is further divided based on the type of cargo being transported. For example, the dry bulk sector focuses on shipping dry commodities such as iron ore, coal, and grain in large quantities.

This thesis studies a Ship Routing and Scheduling (SRS) problem defined in collaboration with Western Bulk, a dry bulk operator. The problem revolves around the optimal deployment of the operator's hired fleet to maximize the overall profit. The operator faces the task of selecting which cargoes to transport from a combination of mandatory contracted cargoes and optional spot cargoes. Costs are incurred for not servicing a contracted cargo. Revenue is generated through the cargo transportation, priced at a unit freight rate measured in USD per metric tonne. The considered variable sailing costs are port expenses, canal fees, and fuel purchases. To maximize profit, the operator must determine routes and schedules for each vessel in its fleet.

In the dry bulk industry, operators often have the flexibility to choose the precise amount of cargo to transport within a given range. This consideration adds complexity to the decision-making process, as determining the optimal amount for each cargo when vessels can carry multiple cargoes simultaneously becomes challenging. Additionally, as ships consume fuel during their voyages, the operator must strategically decide where to refuel and how much bunker (fuel) to purchase. Lastly, Western Bulk aims to distribute its fleet across regions to enhance regional preparedness. Western Bulk employs chartering managers to make such decisions. This study aims to provide chartering managers with an additional tool to support their decision-making process.

In order to represent the operational environment of Western Bulk, this thesis introduces a Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO). The TSRSPBO incorporates several decisions, including cargo selection, determining the quantity of cargo to transport, identifying suitable refueling ports for vessels, optimizing bunker procurement, and allocating vessels to specific regions at the end of the planning period. To effectively model the TSRSPBO, this thesis proposes a two-stage stochastic optimization model. In the first stage, routing and bunkering decisions are addressed, while the second stage incorporates the recourse cost of repositioning the fleet to the dedicated regions.

An arc flow formulation of the TSRSPBO is designed based on previous research by Brønmo et al. (2007b) and Vilhelmsen et al. (2014). The proposed model integrates flexible cargo quantity limits, as explored in Brønmo et al. (2007b), with integrated bunker optimization, as presented by Vilhelmsen et al. (2014). Notably, this study is the first attempt to combine these elements with fleet repositioning. Furthermore, a path flow formulation is introduced, employing a priori column generation as the solution approach. This approach involves generating all feasible vessel routes using a modified Depth-First-Search algorithm. For each route, a linear programming problem is solved using commercial software to optimize cargo quantities and bunker purchases, maximizing the vessel-specific profit. Finally, the path flow model is solved with the generated columns to determine optimal solutions for the TSRSPBO, maximizing the fleet-specific profit.

This thesis generates test instances based on real-life data provided by Western Bulk. Additionally, industry partner Maritime Optima provided accurate routing information. The path flow solution method is finds optimal solutions to test instances of up to 30 cargoes, ten vessels, and ten bunker ports. Western Bulk operates a fleet of more than 110 vessels. Thirty of these are in their long-term portion. Western Bulk reports that their charting managers are doing an excellent job for the remaining vessels in their short- and medium-term fleet. However, for their long-term fleet, Western Bulk acknowledges a potential for improvement (Husby, 2022).

To solve test instances of 30 vessels, an iterative matheuristic solution approach leveraging an Adaptive Large Neighborhood Search (ALNS) framework is proposed. By utilizing the ALNS framework to generate columns, the iterative matheuristic solves the path flow model, called the Vessel Combination Problem (VCP), at regular intervals to obtain high-quality solutions. The resulting matheuristic is named Adaptive Large Neighborhood Search for the Vessel Combination Problem (ALNS-VCP). Results show that the ALNS-VCP finds near-optimal solutions for small- and medium-sized test instances. Additionally, the ALNS-VCP successfully solves test instances of 120 cargoes, 30 vessels, and ten bunker ports in under one hour. As such, the ALNS-VCP outperforms the path flow solution approach with a priori column generation.

This thesis concludes by conducting an economic analysis of Western Bulk's operational environment. The impact of considering fleet repositioning is significant, with a Value of the Stochastic Solution (VSS) of 4.79%. Further, this thesis quantifies the effect of procuring discounted bunker prices at specific ports. Western Bulk devotes resources to procuring bunker purchase contracts. This thesis's results outline the ports on which Western Bulk should focus its efforts.

This thesis is based on the project thesis of Omholt-Jensen (2022) and further extends his work. In particular, although flexible cargo quantities and integrated bunker optimization were investigated in Omholt-Jensen (2022), this thesis incorporates fleet repositioning considerations. To this end, the models are reformulated as two-stage stochastic optimization models. Furthermore, this thesis introduces the iterative ALNS-VCP matheuristic, which can solve realistic test instances.

The remaining chapters in this thesis are organized as follows. Chapter 2 provides an introduction to the industry of maritime transportation. Chapter 3 gives a thorough literature review of relevant research. Chapter 4 describes the TSRSPBO studied in this thesis. Chapter 5 presents the arc flow formulation of the TSRSPBO. Chapter 6 outlines the path flow formulation of the TSRSPBO and describes the a priori column generation approach in detail. Chapter 7 introduces the iterative ALNS-VCP matheuristic leveraged to solve realistic test instances. Chapter 8 describes the construction of test instances from real-life data and provides model implementation details. Chapter 9 conducts a thorough computational study comparing different solution approaches. Chapter 10 summarizes the concluding remarks. Finally, Chapter 11 suggests future research that may be performed based on this thesis.

Chapter 2

Background

This chapter provides an overview of the maritime shipping industry. Specifically, Section 2.1 highlights the role of maritime trade in the global economy. Section 2.2 explains the maritime shipping industry's principles, participants, and trades. Finally, Section 2.3 introduces key characteristics of Western Bulk, an industry partner of this thesis.

2.1 Maritime Transportation and the Global Economy

According to Rodrigue (2020), the maritime transportation industry has become crucial for the global economy in terms of total traded volume due to four key economic factors. Firstly, absolute advantages are attributed to the uneven geographical distribution of resources and the corresponding demand for these resources. To bridge the gap between the vast supply and demand, transportation services are necessary. Given the lack of cost-effective alternatives for long-distance transportation of such large quantities, sea transport is preferred over other modes of transport. Secondly, comparative advantages arise from variations in the production costs and capabilities of manufactured goods, resulting in the trade of these products. As globalization has progressed, trade barriers have diminished, leading to the transportation of assembled products over long distances. Thirdly, advancements in technology have significantly improved the efficiency of maritime transportation. These advancements have been implemented on vessels and cargo-handling terminals on land. Technological solutions have also expanded the range of products transported at sea by adopting highly specialized vessels. Lastly, shipping companies have capitalized on economies of scale by acquiring and utilizing larger vessels in their fleets, thereby increasing cost-effectiveness.

Driven by these influential factors and coinciding with the rise of consumer-driven trends starting in the 1960s, maritime trade experienced substantial growth. From a cargo volume of 500 million metric tonnes in 1950, it has grown by a Compound Annual Growth Rate (CAGR) of approximately 4.5% (Stopford, 2009; UNCTAD, 2022). Consequently, the composition of the global shipping fleet underwent significant changes. During the early 1900s, the fleet primarily consisted of general cargo liners and tramp vessels. General cargo liners followed fixed routes, transporting cargo, passengers, and mail. On the other hand, tramp vessels filled the gaps in the transportation system, carrying various bulk cargoes on the spot market. Both segments were characterized by their diverse cargo types. However, following World War II, the fleet became more specialized due to standardization, automation, economies of scale, and technological advancements (Stopford, 2009). As a result, three significant segments emerged: *bulk shipping*, *specialized shipping*, and *containerization*.

Bulk vessels changed over time, adapting to transporting large quantities of homogeneous cargo between automated terminals for efficient handling. Additionally, the size of these vessels has increased significantly. For instance, the average capacity of tankers, as measured by Deadweight Tonnage (DWT), grew from 16,500 DWT in 1945 to approximately 80,000 DWT in the 1980s. This capacity expansion had a profound impact on reducing unit shipping costs. According to Stopford (2009), US freight prices per tonnage for domestic railway transportation are nearly three times higher than maritime bulk shipments from Japan to the US West Coast.

The concept of specialized cargo carriers emerged to describe vessels specifically designed for the transportation of particular types of cargo. These designs incorporate specific features such as wide hatches for forest product carriers, ramps for Roll-on, Roll-off (Ro-Ro) vessels that transport vehicles, different liquid cargo holds for chemical tankers, pressurized tanks for Liquid Natural Gas (LNG) vessels, and refrigerated holds for Reefer ships that carry frozen goods. Given the high level of control required in the supply chain, companies operating in this sector often adopt vertical integration and prioritize technological innovations.

Lastly, the containerization shipping industry began in 1963 and is responsible for transporting cargo that can be accommodated in standardized containers measuring 40 feet in length, eight feet in width, and eight and a half feet in height. Transportation costs were significantly reduced through extensive integration with rail and trucking services, amounting to approximately 1.5% of the retail value. This reduction represents a remarkable 90% decrease compared to the pre-containerization era (Donovan, 2004).

2.2 Overview of Bulk and Dry Bulk Shipping

This thesis concerns the operation of a dry bulk shipping company. As such, this section provides an overview of the bulk segment, specifically the dry bulk segment.

In 2021, there were approximately 100,000 vessels with a cargo-carrying capacity of over 100 DWT, totaling a capacity of more than 2.1 billion DWT (UNCTAD, 2022). Bulk carriers accounted for 71.8% of the overall capacity among these vessels. These bulk vessels operate under international flags and are strategically designed to optimize tax benefits and regulations. They serve a vast network of 2,916 ports (Aldworth, 2021). The shipbuilding industry comprises 285 shipyards, with over 90% located in China, Japan, or South Korea (Statista, 2022; UNCTAD, 2022). 2019 the dry bulk trade accounted for 5,248 million metric tonnes, predominantly of iron ore, coal, and grain. Similarly, the tanker fleet transported 3,163 million metric tonnes, primarily transporting crude oil and natural gas (UNCTAD, 2022).

The demand for maritime transport primarily comes from multinational corporations that extract raw materials from regions where they are abundant and cost-effective. These corporations utilize vessels to transport large quantities of these raw materials to refining facilities in countries with low labor costs. Refined products are often used as components in other manufacturing processes, leading to further transportation to assembly factories. Bulk shipping plays a crucial role in the initial stage of this supply chain, characterized by large quantities and low unit values.

The Parcel Size Distribution (PSD) is a useful measure for understanding the movement of different cargoes. Stopford (2009) defines a *parcel* as an individual consignment of cargo intended for shipment. The PSD provides insights into the distribution of cargo sizes, which vary depending on the specific type of commodity being traded. The distribution of parcels is influenced by factors such as cargo demand, stock levels, water depth at loading and unloading terminals, and the impact of economies of scale. Commodities with lower values often exhibit PSDs with a higher average value, indicating that they are typically transported in larger quantities. Conversely, high-value commodities are transported in smaller volumes, resulting in PSDs with a lower average value.

Stopford (2009) states that the bulk shipping transport system operates based on four fundamental principles. Firstly, economies of scale play a significant role as larger vessel sizes result in lower unit transportation costs. However, there are limitations to these economies of scale due to the nature of PSD functions (larger vessels would not carry cargo with low PSD values). Additionally, port limitations, such as water depth and vessel size restrictions, can contribute to diminishing returns of the effect of economies of scale. Secondly, an efficient transport system aims to minimize the time spent on cargo handling. Such a system involves investing in high-productivity cargo-handling equipment, such as cranes and grabs for dry bulk cargo or industrial pumps for liquid bulk cargo. Thirdly, the different links within the supply chain should be highly integrated regarding transport modes. This integration can be achieved by strategically located manufacturing plants at coastal sites closely connected to bulk handling terminals, railway stations, and storage facilities. Lastly, optimizing stock holding is crucial for both producers and consumers. This involves considering the impact of holding costs and utilizing inventory theory principles. Just-In-Time (JIT) methodology emphasizes the importance of timed deliveries at the beginning of production, resulting in frequent and smaller cargo loads. This creates a trade-off between optimizing stock holding and taking advantage of economies of scale in transportation. Higher-value commodities are more likely to be shipped in smaller and more frequent voyages.

Bulk shipping companies differentiate themselves along four dimensions: price, speed, reliability, and security (Stopford, 2009). While low transportation costs are generally preferred, the value of the transported cargo often outweighs the importance of transportation costs. Consequently, freight customers are often willing to pay a higher price for transportation services. However, other differentiating factors for high-value commodities may hold greater significance than price alone. Speedy deliveries may be prioritized due to JIT inventory scheduling or significant opportunity costs associated with delays, such as machinery breakdowns. Some customers who follow JIT practices might be willing to pay a premium for a reliable transport service that ensures timely arrivals and minimizes disruptions. Furthermore, freight customers may be willing to pay a higher price for secure transport that guarantees the safe arrival of undamaged goods, especially for high-value products. However, it is essential to note that shipping companies often seek to avoid direct competition along these dimensions by implementing various trade barriers. For instance, they may establish long-term shipping contracts through business relationships or invest in new equipment onboard their vessels to enhance customer attractiveness.

The bulk transport system involves four main participants, as Stopford (2009) describes. The first group consists of *cargo owners* who own global extraction, refining, and manufacturing plants, seeking the lowest labor cost and competitive labor capabilities. Their objective is to secure the most cost-effective transport between their plants. Cargo owners have two options: they can vertically integrate the transport system by acquiring and managing their fleet of vessels, or they can outsource transport through long-term contracts like Contracts of Affreightment (CoA) or charter vessels in the spot market. CoAs are long-term agreements between cargo owners and shipping companies, specifying the volume of cargo to be transported within a given period. Initially, these contracts may have ambiguous terms regarding delivery ports, vessel restrictions, and pickup dates, but they progressively become more specific as the cargo pickup date approaches. The second group comprises *commodity traders* primarily operating in the energy and agriculture markets. They take advantage of arbitrage opportunities by buying cargo in one location and selling it in another. Due to the inherent uncertainty in their trade, commodity traders usually secure transport in the spot market. *Shipowners* form the third group. They own vessels, and their business model revolves around trading vessels based on market conditions. They may order new vessels from shipyards or engage exclusively in the second-hand market. Shipowners often delegate the daily operation of their vessels to specialized companies as they lease their vessels to these operators for fixed or variable payments over a specific period. This arrangement is known as reletting. The final group of participants is the *bulk operators* who act as intermediaries between shipowners and cargo owners. Unlike shipowners, bulk operators typically do not own any vessels themselves. Instead, they manage vessels on behalf of others and operate within the margins between operational costs and revenue generated from transport. Engaging in this line of business entails significant risk. However, a comprehensive understanding of trade patterns, port handling costs, and the trading of freight financial options can help mitigate these risks.

The remainder of this section provides information about the various types of cargo traded in the dry bulk industry, which is relevant to companies like Western Bulk, the industry partner of this thesis. According to UNCTAD (2022), dry bulk carriers accounted for 42.77% of the total capacity of the world fleet in 2021. These carriers are further categorized into subsegments based on their cargo-carrying capacity, as outlined in Table 2.1. Additionally, subsegments often differ in the number of cargo holds they have. A cargo hold is a separate compartment used for transporting cargo, which has a hatch that is opened during cargo loading or unloading and closed during the voyage. The presence of cargo holds enables dry bulk vessels to carry multiple types of cargo simultaneously. Furthermore, vessels are classified into subsegments based on whether they possess self-handling cargo equipment. A vessel capable of loading and unloading cargo is called a *geared* vessel.

Subsegment	**DWT Range (1000s)**	**Geared Equipment**	**Number of Cargo Holds**
Mini-Bulker	0 - 17	Often	Variable
Small Handy	18 - 34	Yes	5
Big Handy	35 - 45	Yes	5
Handymax	46 - 50	Yes	5
Supramax	51 - 58	Yes	5
Ultramax	59 - 66	Yes	5
Panamax	67 - 85	No	7
Post Panamax	86 - 94	No	7
Capesize	95 - 300	No	9

Table 2.1: Different dry bulk segments, and their frequently quoted characteristics. (Husby, 2022)

The products transported in the dry bulk industry can be classified into two main categories. The first category is major bulks, including iron ore, coal, and grain. The second category, minor bulks, encompasses all other dry commodities transported in bulk. This category includes agribulks, sugar, fertilizers, metals & minerals, steel products, and forest products (Stopford, 2009). Table 2.2 provides an overview of the various types of dry bulk cargo and their trade. The numbers presented in the table are compiled from the data provided by UNCOMTRADE (2022), which covers all modes of transportation, not just maritime shipping. Nonetheless, these figures offer some insights into the scale and value of different dry bulk trades.

Iron ore, a vital component in the production of steel, is widely found across the world. However, in 2021, Australia and Brazil emerged as the top two iron ore exporters. This dominance is attributed to their abundant high-grade ore reserves and advantageous coastal mining locations (UNCTAD, 2022).

Coal is a commodity traded in two distinct markets based on its characteristics. Coal with desirable strength and porosity is transformed into coking coal, utilized alongside iron ore in steel production. The remaining types of coal are burned as thermal coal in power stations to generate electricity. Notably, China, India, Japan, and South Korea are the primary importers of coking and thermal coal. Australia primarily exports coking coal, while thermal coal is predominantly supplied by Australia and Indonesia (UNCTAD, 2022).

Grain is crucial in the food industry, utilized directly as human food and indirectly as animal feed for meat production. The grain trade exhibits significant seasonal variations, resulting in volume and trade route fluctuations. Consequently, vessels are primarily chartered from the spot market (Stopford, 2009). Grain availability relies on crop yields and the extent of arable land. Developed nations tend to achieve higher crop yields by leveraging advanced technologies in planting, harvesting, pesticides, and fertilizers. Hence, the leading grain exporters in 2020 were the US, Brazil, Argentina, and Ukraine (UNCTAD, 2022). Income levels and population dynamics influence the demand for grain. As income levels increase, a substitution effect leads to a shift in demand from basic human food towards animal feed, driven by the rising popularity of products like meat and milk. In 2020 the primary grain importers were East and South Asia, Africa, and South and Central America (UNCTAD, 2022).

Agribulk, categorized as a minor bulk, encompasses various agricultural products that do not fall under the grain classification. The main items transported within this category are soya beans and rice. Soya beans are typically processed into soya meals and other vegetable oils, primarily utilized as animal feed. As of 2021, Brazil and the US emerged as the leading exporters, with China being the largest importer (UNCOMTRADE, 2022).

Dry Bulk Cargo Segment	Dry Bulk Cargo Type	Global Export Tonnage (Million Metric Tonnes)	Value of Export (Million USD)
Iron Ore	**Total**	**1,518**	**200,222**
Coal	**Total**	**1,169**	**127,914**
	Coke Coal	1,138	117,645
	Thermal Coal	31	10,269
Grain	**Total**	**439**	**116,396**
	Wheat	188	52,344
	Corn	192	49,241
	Barley	42	10,092
	Grain	11	3,172
	Oats	4	1,015
	Rye	2	532
Agribulks	**Total**	**202**	**97,076**
	Soya Beans	160	76,476
	Rice	39	19,248
	Soya Meal	3	1,352
Sugar	**Total**	**55**	**28,384**
	Raw Sugar	46	21,311
	Bagged Sugar	9	7,073
Fertilizers	**Total**	**98**	**145,443**
	Slag	4	117,645
	Potash	47	13,709
	Urea	29	11,384
	Sulfur	13	2,145
	Phosphate Rock	5	560
Metals & Minerals	**Total**	**516**	**111,625**
	Copper Concentrates	17	68,678
	Petcoke	71	16,322
	Cement & Clinker	157	8,415
	Zinc Concentrates	7	7,962
	Nickel Ore	467	3,195
	Salt	50	2,880
	Alumina & Bauxite	66	2,030
	Gypsum	30	1,006
	Limestone	69	803
	Manganese Ore	3	334
Steel Products	**Total**	**597**	**760,670**
Forest Products	**Total**	**255**	**145,315**

Table 2.2: Types of dry bulk cargo, their global export tonnage and value in 2021 (UNCOMTRADE, 2022).

The sugar dry bulk trades involve the transportation of raw sugar in loose bulk form and refined sugar packaged in parcels. Around 90 countries participate in sugar exportation. Sugar production follows a seasonal pattern with relatively small trading volumes. These characteristics often lead to inadequate loading facilities in many ports that handle sugar. Consequently, the Parcel Size Distributions (PSDs) are low, resulting in the use of smaller vessels for sugar transport. In 2021, Brazil and India held the top positions as the largest exporters, while the US and China ranked as the two largest importers (UNCOMTRADE, 2022).

Agricultural fertilizers primarily consist of chemical elements: nitrogen, potassium, phosphorous, and sulfur. Dry bulk fertilizers commonly include urea, the most prevalent nitrogen-based fertilizer; phosphorous extracted from phosphate rock; muriate of potash, a refined product rich in potassium; slag, a phosphorous fertilizer derived from the steelmaking industry's by-product; and sulfur, which requires special precautions due to its high reactivity, explosive nature, and corrosiveness. When exposed to moisture, sulfur can produce hydrogen sulfide, a dangerous gas (Stopford, 2009). In

2021, Brazil, the US, and India were the largest fertilizer importers, while Russia, China, and the EU were the leading exporters (UNCTAD, 2022)

The trade of metals and minerals encompasses a wide range of products. Alumina, a key component of aluminum production from bauxite, is an intermediate product. Guinea accounted for 46% of the global bauxite supply, with a significant portion exported to China (UNCTAD, 2022). Bauxite, a low-value product with a high PSD, is typically transported in large vessels of Panamax size or larger due to its high volume. On the other hand, refined alumina, which is of higher value and has a lower volume and PSD, is commonly transported in smaller dry bulk vessels. Non-ferrous metals like manganese, copper, nickel, and zinc, which are valuable and have high inventory costs, are often transported in smaller vessels. Regarding construction products, China and the European Union were the top importers of cement and clinkers in 2021, while Turkey and the United Arab Emirates were the leading exporters (UNCOMTRADE, 2022). The largest gypsum importers were the US, India, Japan, and Indonesia, while Oman, the EU, and Spain were the leading exporters (UNCOMTRADE, 2022). Salt was primarily exported from the EU and imported by the US and China.

Steel products are available in various forms, including coils, slabs, bars, billets, pipes, and plates. In 2021, China and the European Union emerged as the largest exporters of steel products, while the United States was the largest importer (UNCOMTRADE, 2022).

Finally, significant exporting countries for the forest products trade include the European Union and Canada, whereas the United States and China are among the importers (UNCOMTRADE, 2022). The forest products trade is characterized by its substantial trade volume and relatively low stowage factor. The *stowage factor* measures density, indicating the volume occupied by a metric tonne of cargo.

2.3 Western Bulk

This section provides an overview of Western Bulk, the industry partner of this thesis, and its relevant operational characteristics. Western Bulk is a dry bulk operator overseeing a fleet of 110 vessels in the Handysize to Supramax dry bulk subsegment, as defined in Table 2.1. The company operates within the profit margins between operational costs and revenue generated from transportation. They are classified as asset-light since they do not possess ownership of the vessels in their fleet. Instead, they acquire vessels through short-term, medium-term, and long-term contracts. Western Bulk is headquartered in Oslo and has additional offices in Singapore, Dubai, Seattle, Casablanca, and Santiago, Chile. With over ten different departments, these teams collaborate to ensure a seamless global operation across the regions depicted in Figure 2.1.

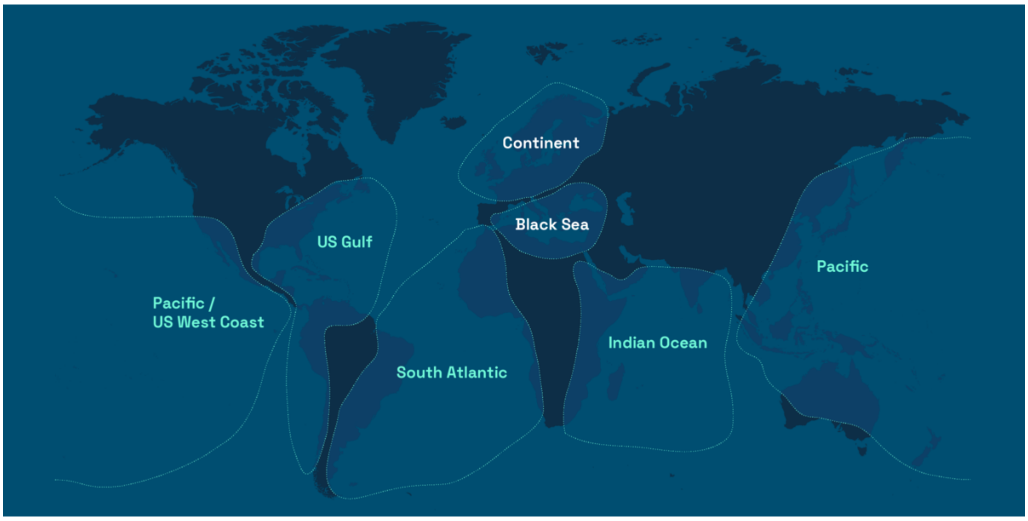


Figure 2.1: The Western Bulk commercial teams' regional responsibilities

The core focus of a dry bulk operator lies in managing a fleet of vessels and maximizing the transported amount of revenue-generating cargo. In the asset-light business model employed by Western Bulk, vessels are chartered or hired under various contract types, including *Time Charter (TC)*, *trip TC*, or *index* contracts.

A TC contract involves the hiring company paying the previous vessel operator a predetermined amount at regular intervals throughout a specified charter period, which has defined start and end dates. Typically, a bulk operator would hire a ship on a TC contract from a shipowner. However, it is also possible for a bulk operator to hire a ship on a TC contract from another bulk operator. In either scenario, the shipowner remains responsible for capital costs and often technical expenses related to crewing, repairs, spare parts, and insurance. On the other hand, the ship's operator assumes responsibility for operational costs associated with the commercial management of the vessel. These costs include bunker fuel, port charges, canal fees, cargo handling fees, cleaning of cargo holds, and other cargo-related expenses.

A trip TC contract can be seen as a variation of a TC contract, specifically tailored to transport a particular cargo within a designated timeframe. In this thesis, a chartered vessel operating under a trip TC contract is called a "spot ship." While regular TC contracts are typically negotiated between shipowners and ship operators, trip TC contracts often involve two ship operator companies as the parties involved. Consequently, trip TC contracts differ slightly from standard TC contracts because certain commercial management costs, including bunkers, port charges, insurance, and canal fees, are covered by the company leasing the ship.

In contrast to TC contracts, index contracts have a distinct characteristic in that the chartering cost is variable and linked to the Baltic Dry Index (BDI). The BDI is a financial index compiled and listed by the Baltic Exchange, a renowned shipping information provider based in London with a long history dating back to the 1700s (BalticExchange, 2022). Each day, a panel of 20 ship broker companies submits the previous day's freight rates for various routes, cargo types, and dry bulk subsegments. Freight rates represent the unit price of transportation and are measured in USD per metric tonne. The emergence of the BDI in the late 1990s was a response to the significant volatility of freight rates in the dry bulk shipping industry (Wilson, 2013). By utilizing the BDI, dry bulk ship operators can employ trading techniques such as hedging and speculation to manage the price fluctuations associated with their industry effectively.

Cargo is available to Western Bulk in the spot market and as part of longer-term Contracts of Affreightment (CoAs). As mentioned in Section 2.2, CoAs specify how much and at which price cargo should be transported for a given period. Historically, Western Bulk services three spot cargoes for every two CoA cargoes. Most vessels in Western Bulk's fleet are chartered on trip TC or TC contracts with durations of less than a year, categorizing them as short- to medium-term contracts. This arrangement results in a relatively high turnover of vessels within their fleet. Additionally, a portion of their fleet is dedicated to vessels on long-term TC contracts spanning over a year. These vessels tend to be among the largest in the fleet, falling within the Supramax to Ultramax subsegments. Western Bulk has maintained a long-term fleet comprising approximately 20-30 vessels throughout history.

The success of a dry bulk operator hinges on its ability to manage portfolio risk effectively. Western Bulk strategically aims to distribute its fleet capacity geographically, ensuring a relatively uniform allocation and promoting preparedness for emerging regional arbitrage opportunities. This approach is essential for the long-term segment of their fleet (Husby, 2022). Regarding cargo composition, Western Bulk maintains a well-diversified portfolio across various dry bulk commodities, as outlined in Table 2.2, with a moderate preference for minerals and coal. Another risk management strategy Western Bulk employs includes parceling, where multiple cargoes are transported simultaneously on the same vessel. Additionally, the company actively engages in financial trading of derivatives linked to the Baltic Dry Index, known as Forward Freight Agreements (FFAs). These agreements serve as contracts between two parties to settle a freight rate for a specified cargo quantity on one of the major dry bulk routes at a predetermined future date. The revenue-generating freight rate is determined when Western Bulk secures a physical cargo in the market. At the time of cargo pickup, the locked-in freight rate might prove to be above or below the FFA curves on the Baltic Dry Index. However, when Western Bulk secured its physical cargo, it may also trade FFA derivatives to hedge or speculate on its secured freight rate. This strategy mitigates the risk of varying freight rates. As an operator, Western Bulk can leverage FFA trading to end up in a relatively risk-neutral market position. In 2021, Western Bulk attributed a gain of 25.8 million USD to using positional FFAs (Western Bulk, 2022).

Western Bulk employs chartering managers to decide which cargoes a vessel should service at what time. They also have to decide the amount of cargo to transport. A chartering manager is often responsible for a fleet of vessels and has to create schedules for one's vessels by having an accurate outlook on the market. Given a belief that certain market conditions are about to arise, one would utilize trading tools to take advantage of a potential arbitrage opportunity. Trading tools might include fixing a spot or CoA cargo and fixing a spot ship, a TC vessel, or an index contract vessel. Fixing, in this context, means contractually securing an asset. Chartering managers may also relet (hire out) vessels on TC, index, or trip TC contracts. Additionally, one might trade FFA contracts to hedge or speculate on future market developments. Finally, a chartering manager might think about the regional allocation of his fleet and collaborate with other internal departments so that the company's fleet is allocated across regions to promote regional preparedness. Finally, once a vessel's schedule has been decided, the operation of the vessel with respect to who should crew the vessel and where to stop for bunker (fuel) must be decided. This thesis denotes the problem faced by the chartering managers of Western Bulk as a Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO), which is further defined in detail in Chapter 4.

Given the unpredictable nature of the dry bulk industry, chartering managers often make decisions based on the reliable information available in the present rather than waiting for potentially better options that may not materialize. Consequently, their decision-making is often focused on achieving locally optimal outcomes rather than considering longer-term planning on a global scale. According to Western Bulk, chartering managers excel in managing their fleet's short-term and medium-term segments. However, they acknowledge the potential for improving long-term fleet management (Husby, 2022). Certain regional arbitrage opportunities may have been missed due to past planning decisions, and some CoA cargoes have been suboptimally planned by utilizing costly spot ships instead of vessels from Western Bulk's long-term fleet. Generally, securing spot ships under time pressure can be expensive since chartering managers have fewer options.

Chapter 3

Literature Review

This chapter provides an overview of the relevant literature on the Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) studied in this thesis. The TSRSPBO is based on the academic field of Ship Routing and Scheduling (SRS), and thus, Section 3.1 gives a high-level summary of research conducted in the SRS field. This thesis introduces three extensions to the traditional SRS problem: a flexible cargo quantities extension, a bunker optimization extension, and a fleet repositioning extension. The literature review in Section 3.2 examines SRS problems that involve flexible cargo quantities. Furthermore, Section 3.3 provides a literature review of bunker optimization extensions in SRS and vehicle routing and scheduling. Additionally, a literature review of other relevant studies is presented in Section 3.4. The resulting literature is synthesized in Section 3.5. Finally, Section 3.6 outlines the contribution of this thesis to the existing literature.

Throughout this thesis, academic literature search engines like *Scopus* (Elsevier, 2022) and *Google Scholar* (Google, 2004) were utilized. Various combinations of search terms such as `ship, vehicle, routing, scheduling, tramp, bunker, reposition,` and `region` were employed to conduct comprehensive searches. Additionally, several review papers were thoroughly examined to gain insights into the existing studies on routing and scheduling problems in maritime and land transportation. Notable review papers that were studied include Ronen (1983, 1993); Christiansen et al. (2004, 2013); Lin et al. (2014); Pache et al. (2019), and Ksciuk et al. (2022). Furthermore, the most highly cited research papers published in the previous year were reviewed to identify any significant mentions of recent literature. Based on the collected literature, two distinct academic avenues were identified. The first pertains to Ship Routing and Scheduling (SRS) problems considering flexible cargo quantities. This field of study has witnessed significant contributions, warranting an in-depth review. The second avenue focuses on routing and scheduling problems incorporating bunker optimization. There were relatively few published articles in the maritime transportation industry addressing these problems, so an exploration of the land transportation industry was also undertaken.

In his review paper, Ronen (1993) provides definitions for commonly used terms in the literature on SRS. The term *shipping* pertains to the transportation of cargo via ships. *Routing* involves determining the order of ports to be visited by a vessel. *Scheduling* incorporates the temporal aspect of routing by assigning specific timestamps to the sequence of port visits. Ronen (1993) additionally categorizes SRS problems into three distinct modes of ship operation: *liner*, *tramp*, and *industrial* operations.

Liner vessels operate according to predetermined routes, exemplified by the container and general cargo shipping industry. On the other hand, *tramp* vessels operate based on available cargoes, seeking opportunities for profit maximization through arbitrage. These cargoes can be contractual (CoA) or optional spot cargoes. Routes are not determined a priori but depend on the availability of cargo. This definition aligns with Stopford (2009)'s definition of bulk operators. Consequently, Western Bulk is classified as a tramp operator in this thesis. *Industrial* operators typically own both the cargo and the vessels used for transportation. They resemble Stopford (2009)'s description of vertically integrated cargo owners. Industrial operators typically focus on cost minimization and consider production levels and storage quantities as decision variables.

Christiansen et al. (2007) categorize maritime transportation planning problems based on their planning horizon, grouping them into three categories. *Strategic* problems have a longer time horizon and involve fleet composition, network design, and transportation system design. *Tactical* problems have a medium-term time horizon, encompassing traditional SRS problems. *Operational* problems involve short-term planning horizon problems, such as selecting cruising speeds and determining environmental routing.

3.1 Ship Routing and Scheduling Literature

The study by Ronen (1983) is a significant and influential contribution to the Ship Routing and Scheduling (SRS) academic discipline. The paper highlights crucial distinctions between conventional vehicle routing and scheduling problems and SRS problems. Ronen (1983) underscores explicitly that vessels possess distinct cost structures, do not always return to their point of origin, operate continuously, and have the potential to alter their destination while at sea.

Significant contributions have been made to the academic field of SRS in recent years, with many summarized in review papers like those by Christiansen et al. (2004) and Christiansen et al. (2013). There are several notable SRS problems studied. Brown et al. (1987) introduce a set partitioning model for SRS problems, specifically for fleets of crude oil tankers with full shiploads. Fagerholt and Christiansen (2000b) utilize dynamic programming to solve a traveling salesperson problem with time windows and pickup and delivery considerations in the context of SRS. Later, Fagerholt and Christiansen (2000a) extend this work to address a multi-ship Pickup and Delivery Problem with Time Windows (PDPTW). Andersson et al. (2011) and Stålhane et al. (2012) investigate a maritime PDPTW with split loads and optional cargoes using exact path-flow models with column generation and a branch-cut-and-price algorithm, respectively.

In addition to exact solution approaches, various heuristic and metaheuristic methods have been applied to SRS problems. Brønmo et al. (2007a) propose a multi-start Local Search, Korsvik et al. (2010) introduce the unified Tabu Search, and Korsvik et al. (2011) and Hemmati et al. (2014) develop a large neighborhood search heuristic. Borthen et al. (2018) employ a hybrid Genetic Algorithm (GA) to solve a multi-period supply vessel planning problem. The Unified Hybrid Genetic Search (UHGS) methodology, previously applied to Vehicle Routing Problems (VRPs), was leveraged by Vidal et al. (2012) and Bulhões et al. (2018). Homsi et al. (2020) extend this methodology to address heterogeneous fixed fleets and customize the Local Search operators for SRS problems. Currently, their work is considered the state-of-the-art solution method in the SRS academic community.

Since the problem investigated in this thesis aligns closely with Ronen (1993)'s definition of tramp operation, a brief overview of various Tramp Ship Routing and Scheduling Problems (TSRSPs) classifications is provided. Pache et al. (2019) argue that there are five significant variations in TSRSPs, including considerations for variable speed, environmental factors, cargo handling variations, bunker optimization, and uncertainties related to sailing time, port duration, demand and supply, spot rate revenue, and weather (Ksciuk et al., 2022).

However, this thesis does not explicitly address variable speed, environmental aspects, or weather-related uncertainties. Furthermore, this thesis assumes fixed parameter values for speed, sailing times, port duration, demand and supply, and spot rate revenues. Noteworthy contributions to speed optimization have been made by Norstad et al. (2011), Gatica and Miranda (2011), Castillo-Villar et al. (2014), and Fan et al. (2019) in their respective papers. Environmental aspects are addressed by Wang et al. (2019) and Li et al. (2022) in their studies. Ksciuk et al. (2022) comprehensively overview various methods to handle uncertainties.

SRS problems in the liner industry typically involve a fixed route and do not usually allow for route deviations, even for refueling purposes (Vilhelmsen et al., 2014). Therefore, this thesis does not offer an extensive overview of the academic literature specifically focused on liner operations.

3.2 SRS Problems with Flexible Cargo Quantities

Pache et al. (2019) provide a comprehensive overview of various cargo constraint extensions explored in the study of Tramp Ship Routing and Scheduling Problems (TSRSPs). Split cargoes, where a single cargo may be serviced by multiple vessels, are investigated by Fagerholt and Ronen (2013) and Lee and Kim (2015). However, as Western Bulk does not operate with split cargoes, these considerations are considered irrelevant to the problem addressed in this thesis. Another interesting cargo constraint extension relates to the concept of cargo coupling, commonly observed in project shipping, where large and bulky cargoes must be transported simultaneously (Fagerholt et al., 2013). In project shipping, vessel operators face the decision of providing transport for either all or none of the cargoes due to limited storage options on board. Therefore, the cargoes are considered coupled. Stålhane et al. (2015) further explore this problem and develop a branch-and-price solution method. While coupled cargoes are relevant to Western Bulk's operational environment, they are not explicitly modeled in the problem studied in this thesis. Hence, this literature review does not include a comprehensive review of coupled cargoes.

An additional cargo constraint extension relevant to the problem addressed in this thesis involves the concept of flexible cargo quantities. In such scenarios, bulk operators have the flexibility to service a variable quantity of cargo within a predetermined interval known as More or Less in Owner's Option (MoLOO) limits. The inclusion of flexible cargo quantity limits provides the tramp operator with the freedom to decide the amount of cargo to transport. As an illustration, consider a scenario where the cargo is specified as 66,000 metric tonnes of thermal coal with a 10% MoLOO. In this case, the MoLOO flexibility limit indicates that the tramp operator can transport any quantity ranging from 59,400 to 72,600 metric tonnes. As MoLOO limits are standard in Western Bulk's operational environment, the cargo constraint extension of flexible cargo quantities is incorporated into the problem studied in this thesis. This section thoroughly reviews the relevant academic literature on TSRSPBOs with flexible cargo quantities.

The mathematical formulation of a TSRSP with flexible cargo quantities was initially presented by Brønmo et al. (2007b) when studying a short-term scheduling problem involving a heterogeneous fleet of vessels. The key distinction from previous mathematical models lies in incorporating cargo quantity as a decision variable. Since the revenue generated is contingent on the transported cargo amount, the problem is formulated as a profit maximization problem. The appeal of operator optionality stems from its ability to transport more cargo with higher freight rates and less cargo with lower freight rates. Moreover, it enables the simultaneous servicing of multiple cargoes. In Brønmo et al. (2007b), a Dantzig-Wolfe decomposition approach transforms the original problem, resulting in a set partitioning formulation. This formulation considers fleet-specific constraints, ensuring that each cargo is exclusively transported by a single vessel and that Contract of Affreightment (CoA) cargoes can be serviced by spot ships. To create all feasible vessel routes adhering to the vessel-specific constraints and compute their respective profits by optimizing the cargo quantity transported on each route, Brønmo et al. (2007b) employ a priori column generation. Subsequently, the set partitioning model selects the subset of feasible vessel routes that maximize fleet profit while respecting fleet-specific constraints. Although the authors were able to solve small test instances to optimality, they conclude that a priori column generation becomes intractable for larger test instances due to the exponential growth in the number of feasible ship routes.

Brønmo et al. (2010) propose a dynamic column generation approach for TSRSPs with flexible cargo quantities to address the computational challenges posed by the exponential growth of feasible routes. In this solution method, columns representing feasible routes are generated dynamically as required rather than generating all feasible routes upfront. The vessel-specific constraints are incorporated into a subproblem, while a restricted master problem is formulated by initially selecting a subset of columns representing feasible routes. Through solving the associated subproblem, additional columns are dynamically added to the master problem. The process continues until the subproblem no longer identifies improving columns. It is important to note that, in their approach, the cargo quantities had to be discretized, rendering the solution method a heuristic. Nonetheless, the solution approach presented by Brønmo et al. (2010) successfully solved larger test instances within a reasonable computational timeframe, yielding acceptable optimality gaps.

Korsvik and Fagerholt (2010) study the same TSRSP problem as Brønmo et al. (2007b) and Brønmo et al. (2010) but implement a Tabu Search heuristic algorithm. The algorithm begins with an initial solution and iteratively moves to the best solution within the current solution's neighborhood. Feasibility checks are conducted for potential moves with cargo quantities set at their lower bounds. Tabu moves are employed to prevent cycling, forbidding solutions resembling recently visited ones. A diversification mechanism reduces the risk of local optima, and periodic reoptimization is performed. Compared to Brønmo et al. (2007b) and Brønmo et al. (2010), Korsvik and Fagerholt (2010) demonstrate faster solving time for the same test cases. The heuristic approach also achieves optimal solutions for problems with known optima. The paper highlights the impact of cargo flexibility limits, showing that a wider range of MoLOO limits leads to nonlinear increases in the profit-maximizing objective function. For instance, a MoLOO range of ± 5% corresponds to approximately a 5% objective value increase, while a 10% ± MoLOO limit results in over a 20% increase.

More recently, a matheuristic-based solution method was proposed by dos Santos et al. (2020) for a maritime cargo routing and shipping problem faced by a chemical company in Brazil. The paper presents a mathematical formulation of a cost-minimizing SRS problem with various considerations, such as multi-product, a heterogeneous fleet with dedicated compartments, draft limits, flexible cargo quantities, split load, and time windows. The developed matheuristic solution method employs a modified relax-and-fix strategy, a relaxation procedure, and a repair and polishing procedure. The relax-and-fit strategy decomposes the planning problem into sequential subproblems solved iteratively. Integer variables are gradually introduced and fixed based on previous iterations until a complete solution is obtained. A post-processing phase, inspired by Rothberg (2007) and Fischetti and Lodi (2008), is applied to improve the best solution found. The matheuristic successfully produced solutions for test instances with up to six origin ports, eight delivery ports, ten different products, and seven vessels. Notably, for smaller instances, the matheuristic outperformed the exact Mixed-Integer Linear Problem (MILP) solver in terms of both solution quality and computational time.

3.3 Routing and Scheduling Problems with Bunker Optimization

Besbes and Savin (2009) conducted one of the early studies focusing on optimal routing and bunker plans for a Ship Routing and Scheduling (SRS) problem. They develop a stochastic dynamic optimization model for a single tramp ship, considering bunkering considerations. Their model incorporates fluctuations in bunker prices using a Markov chain. Notably, their formulation did not consider specific cargo types; instead, they assume that the profit generated from traveling between ports follows a known probability distribution function for each time period. Consequently, their approach aligns more closely with the strategic classification of SRS problems outlined by Christiansen et al. (2007), in contrast to the tactical planning problems with shorter planning horizons. Besbes and Savin (2009) also explore two scenarios for bunker price fluctuations. In the first case, bunker prices could vary over time while remaining consistent across all ports. In the second case, bunker prices could differ across locations but remain fixed throughout the time period. Their model aims to maximize the long-term average profit and does not have an explicit end time. Therefore, it is not suitable for addressing tactical planning problems that involve assigning schedules of cargoes to a fleet of vessels within a predetermined planning horizon.

Vilhelmsen et al. (2014) tackle an SRS problem denoted as a Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) for a fixed fleet of heterogeneous tramp vessels. They introduce bunker price variations, bunker port costs, and time aspects into their model. In their study, bunker consumption depends on a vessel's load, while vessel speeds are fixed at the most economical magnitude. Bunkering can occur at bunker or pickup/delivery ports, but simultaneous bunkering and loading/discharging is not allowed. The authors introduce an arc flow formulation for the TSRSPBO and employ Dantzig-Wolfe decomposition as a solution approach.

They reformulate the problem as a path flow problem, like Brønmo et al. (2010), with fleet-specific constraints in the master problem and vessel-specific constraints in the subproblem. Their solution method involves solving the subproblem using a heuristic shortest path problem with resource constraints algorithm. The optimal route from the subproblem is added as a column in the master problem, and the final solution is obtained by solving the integer version of the master problem. However, the study does not incorporate a branch-and-price algorithm to ensure optimal integral solutions. Test instances with seven vessels, up to 60 cargoes, and 19 bunker options are used to compare the integration of bunker optimization to traditional TSRSPs.

In their paper, Meng et al. (2015) extend the work of Vilhelmsen et al. (2014) on tramp shipping companies with fleets of heterogeneous ships. They propose a branch-and-price framework to obtain solutions with guaranteed optimality gaps. Instead of discretizing bunker amounts, they develop an efficient algorithm that considers real-valued optimal bunkering decisions. They also incorporate cargo demand prediction and adjust daily sailing costs accordingly. Load-dependent bunker consumption and bunker detours are not considered in their approach. Networks generated by Meng et al. (2015) are acyclic, unlike potentially cyclic networks in Vilhelmsen et al. (2014). The branch-and-price approach applies Dantzig-Wolfe decomposition to their arc flow formulation, resulting in a path flow-based master problem. The master problem is solved by choosing a subset of columns for the restricted master problem. The vessel-specific constraints are placed in a subproblem, and the dual variable values are incorporated into the subproblem's objective function. The subproblem is solved using dynamic programming. The optimal subproblem solution is added to the restricted master problem until reduced costs become nonnegative. If the solution is integral, it is the final solution; otherwise, the problem is branched. This approach guarantees optimal solutions without integrality gaps. Meng et al. (2015) demonstrate the effectiveness of their methodology by solving problems for different fleet and cargo sizes, showing improved profit in most instances. Additionally, they compare results with and without including predictions of future cargo demand, showing a slightly higher profit-maximizing objective function value with demand prediction.

The inclusion of refueling policies in the study of Vehicle Routing Problems (VRPs) is relatively recent. Lin et al. (2014) argue that Erdoğan and Miller-Hooks (2012) were the first to do so. In their problem, Erdoğan and Miller-Hooks (2012) consider a heterogeneous fleet of vehicles. Their objective is to minimize total distance while visiting customers and avoiding fuel depletion. They propose a MILP formulation and solve instances using heuristic approaches. Two construction heuristics are employed: the Modified Clarke and Wright Savings heuristic and the Density-Based Clustering Algorithm. The Modified Clarke and Wright Savings algorithm is a modified greedy approach that considers vehicles' fuel capacities. The Density-Based Clustering Algorithm identifies clusters of customers. After initialization, a local neighborhood search with two operators is applied for solution improvement. The best results are provided and compared with the optimal solutions for a set of randomly generated test instances. Their heuristic approach works well compared to exact methods and is shown to solve large problem instances.

In a more recent study, Keskin et al. (2021) formulate an Electric Vehicle Routing Problem with Time Windows (EVRPTW) and stochastic waiting times at recharging stations. In this extended problem, electric vehicles face potential waiting times at recharging stations due to limited chargers available. They argue that waiting times disrupt logistic operations and, as such, aim to produce routes that minimize travel costs and waiting times. To tackle this problem, Keskin et al. (2021) formulate the problem as a two-stage stochastic problem and propose a two-stage simulation-based solution heuristic incorporating an Adaptive Large Neighborhood Search (ALNS) approach. In the first-stage phase, routes are determined based on the expected waiting times at charging stations. The actual waiting times are revealed as the vehicles follow their designated routes and arrive at recharging stations. If the actual waiting times exceed the expected value, the time windows of subsequent customers on the route may be violated. To rectify such infeasible solutions, the second-stage phase penalizes time-windows violations. The authors introduce various destroy and repair operators tailored to address the specific requirements of the problem. Additionally, they propose a novel adaptive mechanism for tuning the constant waiting times used in finding first-stage solutions. To evaluate the performance of their approach and assess the impact of stochastic waiting times on route decisions and costs, Keskin et al. (2021) conduct an experimental study using both small and large test instances from the existing literature. The study results demonstrate that the simulation-based solution approach yields high-quality solutions in terms of both solution quality and computational time, although optimality gaps are not reported. The study highlights the substantial influence that uncertainty in waiting times can have on electric vehicle routes, emphasizing the importance of considering stochastic elements in addressing the EVRPTW.

3.4 Other Relevant Studies

The search for repositioning vessels in the existing academic literature on the Tramp Ship Routing and Scheduling Problem (TSRSP) yielded no relevant results. Most studies regarding repositioning decisions in maritime transportation revolve around the liner industry. Common problems studied include the repositioning of empty containers and liner shipping repositioning of vessels between predetermined routes. This section provides a literature review of the latter, as repositioning empty containers was deemed irrelevant to the problem studied in this thesis.

In their study, Kuhlemann et al. (2021) address a fleet repositioning problem in liner shipping, which involves the costly process of moving container ships with a liner shipping network to adapt to changing customer demands. They highlight a crucial limitation of existing deterministic models for the problem, which overlooks the inherent uncertainty associated with customer demands. Failing to account for this uncertainty can incur additional costs when implementing plans derived from deterministic models. To overcome this limitation, the authors propose a two-stage stochastic optimization model with a binary first stage and a continuous second stage. The study demonstrates the influence of uncertainty on the decisions made in the liner shipping industry. It highlights the advantages of stochastic optimization over deterministic optimization approaches commonly found in the literature. The problem is modeled as a Mixed-Integer Linear Problem (MILP) and solved using Gurobi Optimization, LLC (2022). Kuhlemann et al. (2021) show that by considering uncertainty, the stochastic optimization approach improves the expected performance and enhances the robustness of the repositioning plans, making them more resilient to unpredictable events.

Further, a search for fleet repositioning in land-based routing and scheduling problems primarily resulted in vehicle-sharing problems. For example, He et al. (2020) conducted a study focusing on such a fleet repositioning problem. The objective is to dynamically match vehicle supply and travel demand while minimizing the cost associated with repositioning and lost sales. The authors formulate the fleet repositioning problem as a stochastic dynamic program solved using a distributionally robust optimization framework.

The solution method applied by Ulsrud et al. (2022) was deemed relevant to this thesis. The authors study an operational planning problem in offshore oil and gas. Their objective is to determine optimal routes and sailing speeds for a platform supply vessel fleet to minimize fuel consumption costs. The authors address the challenges of fuel consumption and feasible speed ranges, which are heavily influenced by varying weather conditions. The authors formulate a Time-Dependent Vessel Routing Problem with Speed Optimization (TDVRP-SO). To tackle the computational challenges of solving large-scale instances of the TDVRP-SO using a commercial mixed-integer programming solver, the authors propose an Adaptive Large Neighborhood Search (ALNS) heuristic. The ALNS heuristic is augmented with a local search and a set partitioning model. Computational tests are performed on instances based on a real planning case from the Norwegian continental shelf. The results demonstrate that the ALNS heuristic efficiently produces high-quality solutions.

3.5 Synthesis of the Literature Review

Table 3.1 presents a comprehensive overview and comparison of the papers analyzed in this literature review. These papers can be categorized into three main groups, each addressing distinct aspects of the research domain. The first group, comprising papers 1-4, proposes methods to model and solve SRS problems with flexible cargo quantities. Papers 5-9 belong to the second group and demonstrate various approaches to incorporating bunker decisions into routing and scheduling problems. The third group, represented by papers 10-11, explores handling vehicle repositioning in maritime and land transportation settings. It is worth noting that Paper 12 is not classified within any specific group as its primary significance lies in the authors' implemented solution method rather than its studied problem domain. The problem studied in this thesis encompasses and unifies all three groups by considering a comprehensive Tramp Ship Routing and Scheduling Problem (TSRSP). It addresses the challenges associated with flexible cargo quantities, integrated bunker optimization, and the repositioning of vessels at the end of the planning horizon.

The papers included in the literature review also differ in their objective type. While most TSRSP problems are formulated as profit maximization problems, routing and scheduling problems in VRP and liner literature typically focus on minimizing costs. However, there is an exception. The study of dos Santos et al. (2020) formulates the problem with a cost minimization objective because the authors do not model spot vessels or optional cargoes. Similarly, routing and scheduling problems in VRP and liner transportation literature also lack the option of generating additional revenue. Their main concern is to minimize the costs incurred while satisfying the problem's constraints. These observations influenced the formulation of this thesis' profit maximization objective function. As an operator, Western Bulk often considers optional cargoes based on their profitability and the availability of spare fleet capacity. Therefore, the objective function in this thesis aims to maximize profits by optimizing revenue generation and operational efficiency.

Routing and scheduling studies in maritime transportation differ from those in other transport industries in terms of the presence of central depots and the inclusion of pickup and delivery operations. In the case of VRP problems, the fleet of vehicles starts and ends their routes at a central depot, and pickups are not explicitly considered as they occur at the depot. Similarly, pickup and delivery nodes are not modeled in the context of free-float vehicle-sharing systems studied by He et al. (2020). In contrast, SRS problems do not involve the use of central depots. Vessel operations in SRS problems incorporate pickup and delivery activities, and the fleet has no centralized starting or ending point.

Several of the studies listed in Table 3.1 introduce novel considerations in their mathematical models. For instance, Brønmo et al. (2007b) pioneered the study of a TSRSP that incorporates flexible cargo quantities. Vilhelmsen et al. (2014) present the first TSRSP with integrated bunker optimization. Erdoğan and Miller-Hooks (2012) are the first to consider bunker optimization in the context of VRPs. These studies have played a significant role in shaping the development of the Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) formulated in this thesis. In particular, the model formulation presented in Section 5.3 draws inspiration from the works of Brønmo et al. (2007b) and Vilhelmsen et al. (2014).

A noteworthy aspect of SRS problems is whether a vessel is permitted to transport multiple cargoes simultaneously, referred to as parceling. In the studies focusing on flexible cargo limits, it is generally assumed that the transportation of multiple cargoes is allowed. This assumption is motivated by the potential for higher profits for the vessel operator, as it enables servicing multiple cargoes concurrently. Consequently, this assumption is present in articles 1-4, 10, and 12. However, there are two notable exceptions: the models proposed by Vilhelmsen et al. (2014) and Meng et al. (2015) do not allow for parceling. The study of Vilhelmsen et al. (2014) revolves around a shipping tanker company that primarily operates with full shiploads. Therefore, to simplify their model, they decided to prohibit the transport of multiple cargoes. Similarly, Meng et al. (2015) also restrict cargo transportation to full shiploads with no parceling.

#	Article	Mode of Transportation	Objective	Pickup and Delivery	Time Windows	Optionality of Visiting Nodes	Parceled Cargoes	Flexible Cargo Quantities	Bunker Optimization	Repositioning	Stochastic	Solution Method
1	Brønmo et al. (2007b)	Vessels	Maximize Profit	✓	✓	✓	✓	✓				Set Partitioning and A Priori Column Generation
2	Brønmo et al. (2010)	Vessels	Maximize Profit	✓	✓	✓	✓	✓				Set Partitioning and Heuristic Column Generation
3	Korsvik and Fagerholt (2010)	Vessels	Maximize Profit	✓	✓	✓	✓	✓				Tabu Search Heuristic
4	dos Santos et al. (2020)	Vessels	Minimize Operating Cost	✓	✓		✓	✓				Matheuristic Framework
5	Besbes and Savin (2009)	Vessel	Maximize Profit	✓	✓				✓		✓	Stochastic Dynamic Programming
6	Vilhelmsen et al. (2014)	Vessels	Maximize Profit	✓	✓	✓			✓			Set Partitioning and Heuristic Column Generation
7	Meng et al. (2015)	Vessels	Maximize Profit	✓	✓	✓			✓			Branch-and-Price
8	Erdoğan and Miller-Hooks (2012)	Electric Vehicles	Minimize Travel Distance						✓			Heuristic Search
9	Keskin et al. (2021)	Electric Vehicles	Minimize Cost		✓	✓			✓		✓	Adaptive Large Neighborhood Search
10	Kuhlemann et al. (2021)	Vessels	Minimize Cost	✓	✓		✓			✓	✓	Branch-and-Bound
11	He et al. (2020)	Vehicles	Minimize Cost							✓	✓	Stochastic Dynamic Programming
12	Ulsrud et al. (2022)	Vessels	Minimize Cost	✓	✓	✓	✓					Adaptive Large Neighborhood Search
13	This thesis	Vessels	Maximize Profits	✓	✓	✓	✓	✓	✓	✓	✓	Iterative Matheuristic

Table 3.1: Comparison of important articles and their relation to the TSRSPBO studied in this thesis

There is a limited amount of literature addressing vessel repositioning in the maritime transportation industry. Most studies addressing repositioning primarily focus on the liner industry, making them less applicable to the problem examined in this thesis. However, the study by He et al. (2020) presented an interesting approach to fleet repositioning. In their research, the objective is to distribute a fleet of free-floating vehicles across multiple regions evenly. Although not directly related to the problem studied in this thesis, their work provides a different perspective on fleet repositioning.

Among the studies examined, four of them address the issue of uncertainty. Besbes and Savin (2009) incorporate fluctuations in bunker prices over time and location. Keskin et al. (2021) employ a two-stage stochastic approach, specifically leveraging an Adaptive Large Neighborhood Search (ALNS) method, to minimize waiting times at charging stations in an electric VRP. Kuhlemann et al. (2021) also adopt a two-stage stochastic framework, where routes are determined in the first stage, and the number of transported containers is determined in the second stage. Lastly, He et al. (2020) consider how demand uncertainty across different locations affects the allocation of free-floating vehicles across regions.

The papers examined in this literature review employ various solution methods. Brønmo et al. (2007b) generate all feasible routes and use a set partitioning formulation to obtain exact solutions. Building on this, Brønmo et al. (2010) and Vilhelmsen et al. (2014) incorporate dynamic column generation by solving a subproblem heuristically. Korsvik and Fagerholt (2010) propose a Tabu Search heuristic for their TSRSP with flexible cargo quantities. Besbes and Savin (2009) employ a heuristic dynamic programming approach to maximize expected profits. Meng et al. (2015) develop a branch-and-price framework for exact solutions, restricting bunker detours. Erdoğan and Miller-Hooks (2012) utilize several search heuristics tailored for VRPs, while Keskin et al. (2021) implement a novel two-stage stochastic ALNS framework. Kuhlemann et al. (2021) solve their vessel repositioning problem using an exact commercial solver with branch-and-bound. He et al. (2020) employ a stochastic dynamic programming approach with distributionally robust optimization to find high-quality solutions efficiently. Lastly, Ulsrud et al. (2022) leverage an ALNS framework with local search and set partitioning formulation. Among these solution methods, the approach used by Keskin et al. (2021) and Ulsrud et al. (2022) had the most influence on guiding the solution method adopted in this thesis.

3.6 Our Contribution

To the best of the authors' knowledge, the Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) studied in this thesis is the first problem that combines the elements of a Tramp Ship Routing and Scheduling Problem with the extension of flexible cargo quantities, integrated bunker optimization, and fleet repositioning. This research contributes to the academic field of routing and scheduling problems, specifically enriching and adding complexity to the Tramp Ship Routing and Scheduling Problems. By considering the final destinations of vessels in the fleet at the end of the planning horizon, this thesis extends the work done by Omholt-Jensen (2022) to accurately represent Western Bulk's operational environment. Moreover, an iterative matheuristic approach is proposed, which solves a path flow formulation called the Vessel Combination Problem (VCP) based on columns generated through an Adaptive Large Neighborhood Search (ALNS) procedure. The proposed method, Adaptive Large Neighborhood Search for the Vessel Combination Problem (ALNS-VCP), demonstrates its ability to find high-quality solutions for realistically sized problems. By solving test instances generated from real-life data, the decision-making process of the studied models can provide valuable insights to chartering managers in their professional decision-making processes.

Chapter 4

Problem Definition

This chapter presents the Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) studied in this thesis. The problem is relevant to companies involved in cargo transportation within the dry bulk shipping industry. These companies may consist of dry bulk operators, such as Western Bulk, which manages a fleet of chartered vessels, or integrated ship-owning companies that own the ships in their fleet. Section 4.1 presents the necessary problem input and assumptions, while Section 4.2 outlines the objective, decisions, and restrictions associated with the TSRSPBO.

This thesis provides a comprehensive study of the TSRSPBO, incorporating three notable extensions contributing to its scholarly significance. Firstly, including flexible cargo quantities accurately models the operational dynamics within the dry bulk industry. Secondly, the integration of bunker optimization is incorporated, building upon the findings presented by Vilhelmsen et al. (2014). Their work establishes the added benefits of merging cargo-routing decisions with bunker-related considerations. By this inclusion, the thesis provides a more comprehensive framework for decision-making in the dry bulk industry. Lastly, the introduction of fleet repositioning is motivated by Western Bulk's goal of optimizing its readiness for regional arbitrage opportunities. By incorporating this dimension, the thesis offers valuable insights into enhancing their operational strategies.

Notably, this thesis offers a pioneering effort, as it is the first known study to integrate these extensions within the context of the TSRSPBO. By modeling these extensions, this research contributes to advancing knowledge in the academic field of Ship Routing and Scheduling (SRS).

4.1 Problem Input and Assumptions

The problem inputs and assumptions for the TSRSPBO studied in this thesis can be divided into sections for *vessels, cargoes, ports, regions*, and *sailing legs.*

The studied TSRSPBO considers a tramp operator controlling a heterogeneous fleet of vessels with differing cargo and bunker carrying capacities, sailing speeds, bunker consumption profiles, drafts, and initial bunker levels. A *vessel* is assumed to have a fixed average speed (load-independent) and a specific bunker consumption profile. A vessel's cargo and bunker capacities are given in terms of weight. Moreover, vessels require a minimum water depth level (draft) for safe navigation and may have different cargo handling equipment (gear). These vessel-specific constraints establish feasibility relations between vessels and ports. At the beginning of the planning period, vessels might be located in port or at sea. Additionally, the initial bunker levels are predetermined for each vessel at the start of the planning period. The cost associated with hiring a vessel, known as the charter cost, is defined in the charter contract. The fleet's size remains constant throughout the planning period, and the charter costs for the predefined fleet are considered fixed costs and are not included in the problem formulation. Ships can carry multiple cargoes simultaneously if the total volume and weight do not exceed the vessel's total capacity. However, the specific arrangement of the cargoes within the vessel's cargo holds is not considered.

A predetermined set of *cargoes* is considered to be available throughout the planning period. These cargoes are characterized by their contract type, commodity, weight, flexible quantity limits, pickup ports, delivery ports, pickup and delivery time windows, and freight rate. The cargoes can fall into two categories: Contract of Affreightment (CoA) contracts or optional spot cargoes. Examples of commodities include iron ore, bauxite, or similar items from Table 2.2. Cargoes are picked up in one port and delivered to another. The pickup time windows establish the earliest and latest allowed periods for cargo loading at the pickup port and are considered hard constraints. Similarly, the delivery time windows operate in the same manner. The freight rates are determined through negotiations and are expressed in USD per metric tonne. However, for this problem, it is assumed that the freight rates are known in advance. It is also believed that the cargoes will be available and defined for the entire planning period.

Apart from the predefined fleet, spot ships are available to hire on trip TC contracts to serve CoA cargoes. These vessels are assumed to be available to service any CoA cargo at any given time, with a corresponding charter cost. The charter cost includes any additional trip-related expenses, such as ballast sailing and waiting.

Each *port* is associated with specific cargo handling durations for loading and discharge and specific port costs. Moreover, bunker ports have an assigned bunker price. It is assumed that bunker ports always have a sufficient supply of bunker fuel to refuel a vessel fully and that the bunker prices remain constant over time. Additionally, every port belongs to a designated region. A region represents a grouping of ports situated within a particular geographical area. Each port can only belong to one region and not be part of multiple regions.

A vessel may navigate between pickup ports, delivery ports, bunker ports, or its current position at the beginning of the planning period. It can then proceed to another pickup, delivery, or bunker port from these locations. A *sailing leg* refers to the vessel's direct route between these geographical positions. Each sailing leg is associated with a specific sailing time and cost, which are determined based on factors such as the distance of the sailing leg, the vessel's sailing speed, and its bunker consumption profile. If a ship passes through a canal like the Suez Canal, an additional cost known as a canal cost is incurred. The distance of a sailing leg is assumed to be the shortest sailable geospherical distance. At the start of the planning period, vessels loaded with cargo must deliver their current cargo before being utilized to transport any other cargoes.

In the problem studied in this thesis, it is assumed that cargoes are decoupled, which means that a vessel can transport two different types of cargoes consecutively without any restrictions. Although this assumption simplifies the model, it does not accurately reflect Western Bulk's operational environment. Western Bulk deals with "dirty" cargoes such as salt or cement. After transporting dirty cargo, a vessel would typically incur costs to clean its cargo holds or would be restricted to transporting similar types of cargo. However, these specific considerations are not considered in the problem formulation examined in this thesis.

4.2 Objective, Decisions, and Restrictions

The main objective of the TSRSPBO is to maximize the expected profit for the fleet of vessels during the planning period. Profit is determined by subtracting the total sum of variable sailing costs relevant to this problem from the generated revenue of transported cargo. Each transported cargo unit generates a specific revenue known as the freight rate. Voyage costs, including bunker costs, port costs, and canal costs, as defined by Stopford (2009), are borne by the tramp operator and are considered the variable sailing costs in this problem. Additionally, the problem considers the cost associated with the fleet ending up in unfavorable regions. At the end of the planning period, it is desired to allocate the fleet to regions that promote preparedness for potential regional arbitrage opportunities.

The problem involves making decisions at both the tactical and operational planning levels. The tactical decisions encompass routing decisions, determining whether a spot ship should serve a

CoA cargo, deciding whether a vessel from the fleet should service a spot cargo, and allocating vessels to regions at the end of the planning period. Routing decisions involve assigning specific cargoes to vessels, scheduling cargo pickup and delivery times within designated time windows, and determining the ports and timing for bunker stops. These tactical decisions are interdependent and must be determined simultaneously, as they influence each other.

In this problem, the operational planning decision revolves around determining the quantity of cargo to be transported and the amount of bunker fuel to be purchased. The MoLOO flexibility limits allow the tramp operator to transport any amount within the specified cargo quantity flexibility limits. Therefore, a solution to the TSRSPBO must explicitly specify the precise quantity of cargo to be transported. MoLOO limits enable vessels to maximize their revenue by utilizing their carrying capacity more efficiently. Moreover, the MoLOO flexibility limits increase the possibility of carrying multiple cargoes simultaneously on board a single vessel. A vessel may carry and purchase bunker as long as the carried quantity remains within the bunker capacity of the ship. These modeling decisions necessitate a solution specifying the exact amount of bunker fuel to purchase.

The TSRSPBO examined in this thesis incorporates several restrictions. CoA cargoes must be serviced, while the transportation of spot cargoes is optional. Moreover, all transported cargoes must be picked up within their designated pickup time windows and delivered within their specified delivery time windows. A vessel may be empty but must stay within its capacity limit when transporting cargo. The quantity of cargo loaded at a pickup port must fall within the defined MoLOO flexibility limits. Additionally, for realistic and safe operation, a vessel's bunker level must be above a safety limit and below its bunker capacity limit. Vessels cannot service cargoes in ports with a water depth lower than their draft limit. Furthermore, vessels must comply with the cargo handling restrictions specific to each port. Finally, at the end of the planning period, the allocation of vessels in each region should be strategically planned to ensure preparedness in the event of regional arbitrage opportunities.

Chapter 5

Mathematical Arc Flow Formulation

This section presents a mathematical arc flow formulation for the Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) investigated in this thesis. Section 5.1 discusses the design choices that lead to a two-stage stochastic optimization model. The notation used in the mathematical model is defined in Section 5.2. The complete mathematical arc flow model studied in this thesis is presented in Section 5.3. Section 5.4 elaborates on the linearizations necessary to solve the model using commercial Mixed-Integer Linear Programming (MILP) solvers.

5.1 Modeling Approach and Assumptions

Chapter 4 states that cargoes are assigned a delivery time window. However, cargo quotes provided to the tramp operator only include a pickup time. The purchaser of transport is guaranteed cargo delivery through a contractual document known as a Bill of Lading. However, the Bill of Lading specifies that delivery should occur at the earliest convenience or at similarly ambiguous timeframes. To avoid the possibility of a vessel continuously carrying a small cargo while servicing more profitable cargoes, artificial delivery time windows were introduced. These artificial time windows are created by multiplying the sailing and port times by a specific factor (e.g., 2). This design decision allows for deviations to purchase bunker fuel and pick up other cargoes while ensuring that all cargoes are delivered within a reasonable time.

Additionally, this thesis assumes that the time required to purchase and load bunker fuel is independent of the quantity of cargo being loaded. This assumption aligns with Vilhelmsen et al. (2014), where the fixed time cost associated with a vessel berthing and preparing to take on bunker fuel typically outweighs the variable time spent on loading the bunker fuel itself. Consequently, a fixed amount of time for bunker operations is added to the sailing time for trips that involve arriving at a bunker port.

At the beginning of the planning period, vessels may be located either at a port or at sea. As they operate continuously throughout the day, it is unlikely for all vessels to be in port simultaneously. Therefore, vessels become available at specific ports, corresponding to the discharge port of a vessel's ongoing voyage. Upon completing its service at the last planned discharge port, a vessel concludes its route by traveling to an artificial destination node at zero cost and time. This destination node is assumed to be within the same region as the preceding delivery or bunker node. The amount of bunker fuel carried by vessels at the destination node is considered a resource and is priced accordingly. The difference in bunker load between the origin and destination nodes is valued based on the average price of all bunker ports. The value of the additional bunker is reflected in the model's objective function.

Some bunker ports can also serve as pickup or delivery ports for cargo. This duality means that there is a possibility of simultaneous bunkering and cargo-loading activities. However, such concurrent operations are not permitted in the TSRSPBO studied in this thesis. Instead, for trips between ports that involve both cargo-related and bunker-related activities, the travel time, cost, and bunker consumption are all set to zero. This consideration allows for subsequent but not simultaneous loading of cargo and bunkering activities. Consultations with Western Bulk have confirmed that excluding the option of concurrent bunkering is reasonable. It was found that less than 20% of their port visits involved this type of bunkering activity (Husby, 2022).

In theory, vessels can travel from one bunker node to another. This scenario could materialize when a vessel with low fuel levels stops at an expensive bunker port to partially refill its bunker tank before proceeding to a more cost-effective bunker port for a complete refill. However, Husby (2022) indicated that such scenarios are rare. As incorporating such bunker-to-bunker legs would significantly increase the complexity of the model, they are excluded.

As the TSRSPBO studied in this thesis optimizes the expected fleet-wide profit during the planning horizon, optimal solutions should be characterized by favorable vessel distributions across regions. To model such characteristics, constraints are introduced that determine the number of vessels allocated in each region at the end of the planning period. These constraints reflect the ship operator's market expectations, with more vessels allocated to regions expected to offer numerous profitable cargoes. Conversely, fewer vessels are allocated to regions with worse outlooks. A repositioning cost is introduced to assess the quality of regional vessel allocations. This vessel-specific cost considers sailing time, fuel consumption, and potential canal fees and estimates the cost of repositioning a vessel from its final destination to a given region. The profit-maximizing objective function of the model is extended by subtracting the total repositioning cost required to adhere to the regional constraints. The repositioning term penalizes unfavorable allocations and rewards favorable ones. The resulting model becomes a two-stage optimization model, where routing decisions are decided in the first-stage, and the recourse cost of repositioning the fleet is calculated in the second-stage. It is important to note that second-stage cost is solely used to price regional vessel distributions and does not provide operational support. Due to the uncertainty regarding the number of vessels to be allocated in each region, stochastic scenarios are introduced with associated probabilities. Thus, regional constraints are defined for each region and scenario, resulting in a stochastic two-stage optimization model that is leveraged to model the TSRSPBO studied in this thesis.

5.2 Arc Flow Notation

The notation and formulation presented in this section follow the arc flow formulation of the Tramp Ship Routing and Scheduling Problem presented by Christiansen et al. (2007), the flexible cargo quantities extension formulated in Christiansen and Fagerholt (2014), and the bunker management extension devised by Vilhelmsen et al. (2014). Additional notation is introduced to present the stochastic formulation of the problem.

Let the fleet of vessels be represented by the set $\mathcal{V}$, indexed by v. Assume N cargoes are available for transport, indexed by i. The sets of pickup and delivery nodes are given by $\mathcal{N}^P = \{0, \ldots, N\}$ and $\mathcal{N}^D = \{N+1, \ldots, 2N\}$, respectively. These sets allow each cargo i to be represented by a pickup node $i \in \mathcal{N}^P$ and a delivery node $N+i \in \mathcal{N}^D$. As such, the set of all cargo-related nodes is represented as $\mathcal{N} = \mathcal{N}^P \cup \mathcal{N}^D$. The set of pickup nodes $\mathcal{N}^P$ is partitioned into $\mathcal{N}^P = \mathcal{N}^C \cup \mathcal{N}^O$, where $\mathcal{N}^C$ and $\mathcal{N}^O$ denote the mandatory contracted and optional spot cargoes, respectively. A vessel has a node of origin $o(v)$, a geographical position at sea or in port. It is assigned an artificial destination node $d(v)$, which a vessel will travel to after its last planned discharge node service. Due to cargo capacity, depth, and cargo handling equipment restrictions, some vessels and nodes in $\mathcal{N}$ may be incompatible. Thus, the set of nodes a vessel v may visit is represented as $\mathcal{N}_v \subseteq \mathcal{N} \cup \{o(v), d(v)\}$. $(\mathcal{N}_v, \mathcal{A}_v)$ defines a base network for each vessel v, where the set of arcs $\mathcal{A}_v \subseteq \{(i,j) | i \in \mathcal{N}_v, j \in \mathcal{N}_v\}$ represents all arcs traversable by vessel v with respect to time and bunker consumption. The pickup and delivery nodes compatible with vessel v are defined as $\mathcal{N}_v^P = \mathcal{N}^P \cap \mathcal{N}_v$ and $\mathcal{N}_v^D = \mathcal{N}^D \cap \mathcal{N}_v$ for convenience.

The next step is extending the base network $(\mathcal{N}_v, \mathcal{A}_v)$ to allow vessel visits at bunker option nodes. Let $\mathcal{B}$ be the set of bunker option nodes. Similarly, as for the cargo-related nodes, vessels and bunker option nodes may be incompatible. Thus, set $\mathcal{B}_v \subseteq \mathcal{B}$ denotes the bunker option nodes that vessel v may visit. The set of base nodes $\mathcal{N}_v$ for vessel v is extended to include the nodes in $\mathcal{B}_v$ such that $\hat{\mathcal{N}_v} = \mathcal{N}_v \cup \mathcal{B}_v$. Further, the set of base arcs $\mathcal{A}_v$ is extended by adding all arcs connecting nodes in $\mathcal{N}_v \setminus d(v)$ with nodes in $\mathcal{B}_v$ that are traversable by vessel v with respect to time and bunker. The arcs from the destination node $d(v)$ are excluded as the next node to visit is unknown. As such, the extended set of arcs is defined as $\hat{\mathcal{A}}_v = \mathcal{A}_v \cup \mathcal{A}_v^B$, where $\mathcal{A}_v^B$ is the set of arcs for vessel v connecting bunker option nodes $\mathcal{B}_v$ to the nodes in $\mathcal{N}_v \setminus d(v)$. The extended cargo-bunker network is thereby $(\hat{\mathcal{N}_v}, \hat{\mathcal{A}}_v)$.

Each node in the $(\hat{\mathcal{N}_v}, \hat{\mathcal{A}}_v)$ graphs is associated with a region $k \in \mathcal{K}$. The set of scenarios is represented by $\mathcal{S}$, indexed by s.

Let T_{ijv}^S be the sailing time from node i to node j for a given vessel v and arc $(i,j) \in \hat{\mathcal{A}}_v$. If node j is a bunker node, a fixed time is added to T_{ijv}^S, signifying the time spent bunkering. The time required to load or discharge one unit of cargo i with vessel v is defined as T_{iv}^Q. The variable voyage cost C_{ijv} accounts for the costs of visiting node i and the sailing costs from node i to node j for vessel v. It is important to note that the cost of the purchased bunker is not included in this parameter as the purchase of bunker will be modeled separately. Instead, let P_i^B denote the cost of purchasing one unit of bunker available in bunker option node $i \in \mathcal{B}$. The bonus bunker remaining at the destination node $d(v)$ is represented as $\widetilde{P}$ and calculated as the mean of all available bunker node prices. The charter cost of servicing a contracted cargo i by a spot ship is denoted C_i^S. Further, the bunker consumption at sea is denoted B_{ijv}^S for a vessel v traversing the arc $(i,j) \in \hat{\mathcal{A}}_v$. The bunker consumed by vessel v in node i per unit of time is represented by B_{iv}^P. There is a unit revenue R_i generated for transporting cargo $i \in \mathcal{N}^P$. For each node $i \in \hat{\mathcal{A}}_v$ visited by vessel v, there is a time window $\left[\underline{T}_{iv}, \overline{T}_{iv}\right]$ for the pickup and delivery time, defined by the earliest time $\underline{T}_{iv}$ and the latest time $\overline{T}_{iv}$ of the visit. Additionally, the bunker level on board each vessel v must be within a lower limit $\underline{B}_v$ and an upper limit $\overline{B}_v$ to ensure realistic and safe operation. The quantity of cargo that can be transported is flexible within interval $\left[\underline{Q}_i, \overline{Q}_i\right]$, where $\underline{Q}_i$ is the minimum quantity and $\overline{Q}_i$ is the maximum quantity that must be transported if cargo i is serviced. The cargo carrying capacity of vessel v is denoted K_v. Let A_{ik} be equal to 1 if node i is in region k, and 0 otherwise. Let R_{ks}^K represent the number of vessels that are to be allocated in each region, r, and in each scenario, s. Further, let $C_{d(v)k}^B$ represent the repositioning cost for vessel, v, from the last visited node, $d(v)$, to region k. Finally, let P_s represent the probability of scenario s.

The binary variable x_{ijv} is assigned the value 1 if vessel v traverses the arc $(i,j) \in \hat{\mathcal{A}}_v$, and 0 otherwise. The variables t_{iv} denote the time vessel v begins service at node i. The cargo load on board a vessel v when it leaves node i is represented by the variable l_{iv}^C. Similarly, the variable l_{iv}^B represents the bunker load on board vessel v just after completing service at node i. The quantity of cargo i transported by vessel v is represented by q_{iv}, and the quantity of bunker purchased by vessel v at bunker option node i is given by the variable b_{iv}. To represent whether a contracted cargo is serviced by a spot ship, the binary variable z_i is assigned the value 1, or 0 otherwise. The binary variable y_i is assigned a value of 1 if the optional spot cargo i is serviced, and 0 otherwise. Let the binary variable r_{vk} be equal to 1 if vessel v's last visited node is located in region k, and 0 otherwise. Finally, let the binary stochastic second-stage variable $x_{d(v)vks}^B$ be equal to 1 if vessel v repositions from its last visited node $d(v)$ to region k in scenario s, and 0 otherwise.

Tables 5.1 - 5.2 provide a summary of the sets, variable, and parameters introduced in this section, respectively.

Set Notation	Set Description
$\mathcal{V}$	Set of vessels
$\mathcal{B}$	Set of bunker option nodes
$\mathcal{B}_v$	Set of bunker option nodes that vessel v may service
$\mathcal{N}$	Set of all cargo-related nodes
$\mathcal{N}^D$	Set of delivery nodes
$\mathcal{N}^P$	Set of pickup nodes
$\mathcal{N}^C$	Set of pickup nodes for the mandatory contracted cargoes
$\mathcal{N}^O$	Set of pickup nodes for the optional spot cargoes
$\mathcal{N}_v$	Set of nodes that can be serviced by vessel v, including the origin node and an artificial destination node
$\mathcal{N}_v^P$	Set of pickup nodes that vessel v may service
$\mathcal{N}_v^D$	Set of delivery nodes that vessel v may service
$\mathcal{A}_v$	Set of all arcs traversable by vessel v
$\hat{\mathcal{N}_v}$	Extended set of nodes and bunker option nodes that vessel v may service
$\hat{\mathcal{A}}_v$	Extended set of arcs between nodes and bunker option nodes that vessel v may traverse
$\mathcal{K}$	Set of shipping regions
$\mathcal{S}$	Set of scenarios

Table 5.1: Model Sets

Variable Notation	Variable Domain	Variable Description
x_{ijv}	$v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v$	1 if vessel v services node i just before node j, 0 otherwise
t_{iv}	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}$	Time of which vessel v begins service at node i
b_{iv}	$v \in \mathcal{V}, i \in \mathcal{B}_v$	The bunker quantity purchased by vessel v at node i
l_{iv}^C	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}$	Cargo load on board vessel v just after completing service at node i
l_{iv}^B	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}$	Bunker load on board vessel v just after completing service at node i
q_{iv}	$v \in \mathcal{V}, i \in \mathcal{N}_v^P$	Quantity of cargo i that is transported by vessel v
y_i	$i \in \mathcal{N}^O$	1 if the optional spot cargo is transported, 0 otherwise
z_i	$i \in \mathcal{N}^P$	1 if cargo at node i is serviced by vessel from the spot market, 0 otherwise
r_{vk}	$v \in \mathcal{V}, k \in \mathcal{K}$	1 if vessel v's last visited node is located in region k, 0 otherwise
$x_{d(v)vks}^B$	$v \in \mathcal{V}, k \in \mathcal{K}, s \in \mathcal{S}$	1 if vessel v repositions from its last visited node to region k in scenario s, 0 otherwise

Table 5.2: Model Variables

Parameter Notation	Parameter Domain	Parameter Description
N		Number of available cargoes
T^S_{ijv}	$v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v$	Sailing time from node i directly to node j for vessel v
T^Q_{iv}	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}$	Time required to load or discharge one unit of cargo at node i with vessel v
C_{ijv}	$v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v$	Cost of servicing node i and sailing directly from node i to node j with vessel v, cost of purchasing bunker not included
P^B_i	$i \in \mathcal{B}$	Price of purchasing one unit of bunker at the bunker option i
$\widetilde{P}$		Unit price of bonus bunker
C^S_i	$i \in \mathcal{N}^P$	Cost of servicing the cargo at node i with a spot ship
B^S_{ijv}	$v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v$	Total bunker consumption for vessel v while sailing directly from node i to node j
B^P_{iv}	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}$	Port bunker consumption for vessel v while in node i
B^0_v	$v \in \mathcal{V}$	Initial bunker level on board vessel v
R_i	$i \in \mathcal{N}^P$	Revenue generated from transporting one unit of cargo from node i
$\overline{T}_{iv}$	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}$	The latest time at which vessel v may begin its service at node i
$\underline{T}_{iv}$	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}$	The earliest time at which vessel v may begin its service at node i
$\overline{B}_v$	$v \in \mathcal{V}$	Maximum bunker level for vessel v
$\underline{B}_v$	$v \in \mathcal{V}$	Minimum bunker level for vessel v
$\overline{Q}_i$	$i \in \mathcal{N}^P$	Maximum quantity of the cargo at node i to be transported
$\underline{Q}_i$	$i \in \mathcal{N}^P$	Minimum quantity of the cargo at node i to be transported
K_v	$v \in \mathcal{V}$	Cargo carrying capacity of vessel v
A_{ik}	$i \in \hat{\mathcal{N}_v}, k \in \mathcal{K}$	1 if node i is in region k, 0 otherwise
R^K_{ks}	$k \in \mathcal{K}, s \in \mathcal{S}$	Number of vessels to be allocated in region k in scenario s
$C^B_{d(v)k}$	$v \in \mathcal{V}, k \in \mathcal{K}$	Reposition cost from vessel v's last visited node to region k
P_s	$s \in \mathcal{S}$	Probability of scenario s

Table 5.3: Model Parameters

5.3 Arc Flow Mathematical Model

This section provides a mathematical model of the stochastic two-stage TSRSPBO studied in this thesis. In particular, Section 5.3.1 provides the first-stage formulation of the problem. Section 5.3.2 presents the second-stage phase of the problem.

5.3.1 Arc Flow First-Stage Formulation

This section provides the first-stage formulation of the two-stage TSRSPBO, including the first-stage objective function, network flow constraints, temporal constraints, cargo constraints, bunker constraints, regional constraints, and variable domain constraints.

Arc Flow Objective Function

$$\begin{aligned}\max \quad & \sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{N}^P} R_i q_{iv} - \sum_{v \in \mathcal{V}} \sum_{(i,j) \in \hat{\mathcal{A}}_v} C_{ijv} x_{ijv} - \sum_{i \in \mathcal{N}^C} C_i^S z_i \\ & - \sum_{v \in \mathcal{V}} \sum_{i \in \mathcal{B}_v} P_i b_{iv} + \sum_{v \in \mathcal{V}} \widetilde{P} \cdot (l_{d(v)v}^B - B_v^0) - \mathcal{Q}(\boldsymbol{r})\end{aligned} \tag{5.1}$$

The objective function (5.1) aims to maximize the total profit for the deployed fleet. The first term specifies the revenue generated for both contracted and optional cargoes as a function of the quantity transported. The second term specifies the variable voyage costs associated with servicing a cargo. The final term on the first line represents the charter costs associated with servicing a contracted cargo with a spot ship. Together, the first line describes the objective presented by Christiansen and Fagerholt (2014). The fourth term describes the cost associated with purchasing bunker, and the fifth term models the value of the bonus bunker remaining at the destination node. Note that if the bunker level at the destination is higher than at the origin, this term will positively contribute to the objective. Conversely, a lower bunker level at the destination than at the origin yields a negative contribution. The final $\mathcal{Q}(\boldsymbol{r})$ term denotes the second-stage recourse cost for repositioning vessels to suitable regions. The symbol $\boldsymbol{r}$ denotes the collection of r_{vk} variables stating whether a vessel v's last visited node is located in region k, or not. The $\mathcal{Q}(\boldsymbol{r})$ term is explained further in Section 5.3.2.

Arc Flow Network Flow Constraints

$$\sum_{v \in \mathcal{V}} \sum_{j \in \hat{\mathcal{N}}_v} x_{ijv} + z_i = 1 \qquad i \in \mathcal{N}^C \tag{5.2}$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in \hat{\mathcal{N}}_v} x_{ijv} - y_i = 0 \qquad i \in \mathcal{N}^O \tag{5.3}$$

$$\sum_{j \in \hat{\mathcal{N}}_v} x_{o(v)jv} = 1 \qquad v \in \mathcal{V} \tag{5.4}$$

$$\sum_{j \in \hat{\mathcal{N}}_v} x_{ijv} - \sum_{j \in \hat{\mathcal{N}}_v} x_{jiv} = 0 \qquad v \in \mathcal{V}, i \in \hat{\mathcal{N}}_v \setminus \{o(v), d(v)\} \tag{5.5}$$

$$\sum_{i \in \hat{\mathcal{N}}_v} x_{id(v)v} = 1 \qquad v \in \mathcal{V} \tag{5.6}$$

$$\sum_{j \in \hat{\mathcal{N}}_v} x_{ijv} - \sum_{j \in \hat{\mathcal{N}}_v} x_{N+i,jv} = 0 \qquad v \in \mathcal{V}, i \in \mathcal{N}_v^P \tag{5.7}$$

Constraints (5.2) ensure that all contract cargoes are transported by a vessel in the predefined fleet or by a spot ship. In contrast, Constraints (5.3) state that all spot cargoes must be transported by a vessel in the predefined fleet or not at all. Constraints (5.4) ensure that all vessels begin a voyage from their origin node and travel to a pickup node, a bunker option node, or directly to their destination node. Constraints (5.5) denote the flow conservation, while Constraints (5.6) ensure that all vessels travel to their destination node, either from a delivery node, a bunker option node, or their origin node. Together, Constraints (5.4) - (5.6) describe the flow on the sailing route used by vessel v. For each cargo i and vessel v, Constraints (5.7) ensure that the same vessel services both the pickup and the cargo's corresponding delivery node.

Arc Flow Temporal Constraints

$$x_{ijv}\left(t_{iv}+T^{Q}_{iv}q_{iv}+T^{S}_{ijv}-t_{jv}\right)\leq 0 \qquad v\in\mathcal{V},(i,j)\in\hat{\mathcal{A}}_v \;|i\in\mathcal{N}^{P}_{v} \tag{5.8}$$

$$x_{ijv}\left(t_{iv}+T^{Q}_{iv}q_{i-N,v}+T^{S}_{ijv}-t_{jv}\right)\leq 0 \qquad v\in\mathcal{V},(i,j)\in\hat{\mathcal{A}}_v \;|i\in\mathcal{N}^{D}_{v} \tag{5.9}$$

$$x_{ijv}\left(t_{iv}+T^{S}_{ijv}-t_{jv}\right)\leq 0 \qquad v\in\mathcal{V},(i,j)\in\hat{\mathcal{A}}_v \;|i\notin\mathcal{N}^{P}_{v}\wedge i\notin\mathcal{N}^{D}_{v} \tag{5.10}$$

$$t_{iv}+T^{Q}_{iv}q_{iv}+T^{S}_{i,N+i,v}-t_{N+i,v}\leq 0 \qquad v\in\mathcal{V},i\in\mathcal{N}^{P}_{v} \tag{5.11}$$

$$\underline{T}_{iv}\leq t_{iv}\leq\overline{T}_{iv} \qquad v\in\mathcal{V},i\in\hat{\mathcal{N}}_v \tag{5.12}$$

The above constraints handle the temporal aspects of the problem. Constraints (5.8) state that if vessel v travels directly from node i to node j, then, the time at which a vessel v begins loading of cargo in node j must be greater than or equal to the time at which the vessel began loading of cargo in node i plus the sum of a quantity dependent loading time at node i and the sailing time from node i to node j. Similarly, Constraints (5.9) specify the time progression for delivery nodes if vessel v travel directly from node i to node j. Note, however, that the cargo quantity variable q_{iv} is only defined for pickup nodes, and as such the term that specifies the variable discharge time uses a different index. Constraints (5.10) describe a similar time flow for origin, bunker, and destination nodes as there is no cargo loading or discharging at these nodes. Constraints (5.11) are precedence constraints forcing the pickup node to be serviced before its corresponding delivery node. The inequality signs of Constraints (5.8) - (5.11) give vessels the option to wait before beginning their service at node j. Finally, Constraints (5.12) are the time window constraints of within a vessel must begin their service at node i.

Arc Flow Cargo Constraints

$$x_{ijv}\left(l^{C}_{iv}+q_{jv}-l^{C}_{jv}\right)=0 \qquad v\in\mathcal{V},(i,j)\in\hat{\mathcal{A}}_v \;|j\in\mathcal{N}^{P}_{v} \tag{5.13}$$

$$x_{i,N+j,v}\left(l^{C}_{iv}-q_{jv}-l^{C}_{N+j,v}\right)=0 \qquad v\in\mathcal{V},(i,N+j)\in\hat{\mathcal{A}}_v \;|j\in\mathcal{N}^{P}_{v} \tag{5.14}$$

$$x_{ijv}\left(l^{C}_{iv}-l^{C}_{jv}\right)=0 \qquad v\in\mathcal{V},(i,j)\in\hat{\mathcal{A}}_v \;|j\in\mathcal{B}_v \tag{5.15}$$

$$l^{C}_{o(v),v}=0 \qquad v\in\mathcal{V} \tag{5.16}$$

$$q_{iv}\leq l^{C}_{iv}\leq\sum_{j\in\hat{\mathcal{N}}_v}K_v x_{ijv} \qquad v\in\mathcal{V},i\in\mathcal{N}^{P}_{v} \tag{5.17}$$

$$0\leq l^{C}_{N+i,v}\leq\sum_{j\in\hat{\mathcal{N}}_v}\left(K_v-q_{iv}\right)x_{N+i,jv} \qquad v\in\mathcal{V},i\in\mathcal{N}^{P}_{v} \tag{5.18}$$

$$\sum_{j\in\hat{\mathcal{N}}_v}\underline{Q}_i x_{ijv}\leq q_{iv}\leq\sum_{j\in\hat{\mathcal{N}}_v}\overline{Q}_i x_{ijv} \qquad v\in\mathcal{V},i\in\mathcal{N}^{P}_{v} \tag{5.19}$$

The constraints in this paragraph describe the restrictions related to the handling of the cargo transported. Constraints (5.13) state that if a vessel v travels directly from node i to a pickup node j, then, the cargo load on board the vessel just after completing service at node i plus the cargo loaded at node j must be equal to the cargo level on board the vessel just after completing service at node j. Constraints (5.14) specify a similar cargo flow if vessel v travels directly from node i to delivery node $N + j$. Further, Constraints (5.15) enforce that if a vessel v travels to a bunker node j, then the cargo load on board the vessel stays the same as there is no cargo loading or discharging at bunker nodes. Note that the equality sign in Constraints (5.13) - (5.15) ensure that no cargo is misplaced during the shipment. Constraints (5.17) describe the cargo load capacity restrictions at pickup nodes, i.e., that the cargo loaded by vessel v at pickup node i must be less or equal to the cargo load on board just after the vessel completes its service of node i which must be less or equal to the total cargo carrying capacity of the vessel. Constraints (5.18) specify the cargo discharge capacity restrictions at delivery nodes, i.e., that the cargo load on board vessel v just after completing its service at delivery node $N + i$ must be greater or equal to zero and less than or equal to the cargo load capacity of the vessel less the cargo loaded at pickup node i. Finally, Constraints (5.19) ensure that the cargo loaded at pickup port i is within the minimum and maximum cargo load limits. The summations in Constraints (5.17), (5.18), and (5.19) are introduced to strengthen the formulation of the model by enforcing only one of the x_{ijv} variables involved in the summation to be non-zero.

Arc Flow Bunker Constraints

$$x_{ijv}\left(l^B_{iv} - B^S_{ijv} - B^P_{jv}T^Q_{jv}q_{jv} - l^B_{jv}\right) = 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; |j \in \mathcal{N}^P_v \tag{5.20}$$

$$x_{ijv}\left(l^B_{iv} - B^S_{ijv} - B^P_{iv}T^Q_{jv}q_{j-N,v} - l^B_{jv}\right) = 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; |j \in \mathcal{N}^D_v \tag{5.21}$$

$$x_{ijv}\left(l^B_{iv} - B^S_{ijv} + b_{jv} - l^B_{jv}\right) = 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; |j \in \mathcal{B}_v \tag{5.22}$$

$$x_{ijv}\left(l^B_{iv} - B^S_{ijv} - l^B_{jv}\right) = 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; |j \in \{d(v)\} \tag{5.23}$$

$$l^B_{o(v)v} = B^0_v \qquad v \in \mathcal{V} \tag{5.24}$$

$$\sum_{j \in \hat{\mathcal{N}}_v} \left(\underline{B}_v + B^S_{ijv} + B^P_{jv}T^Q_{jv}q_{jv}\right) x_{ijv} \le l^B_{iv} \le \overline{B}_v \sum_{j \in \hat{\mathcal{N}}_v} x_{ijv} \qquad v \in \mathcal{V}, i \in \hat{\mathcal{N}}_v \tag{5.25}$$

$$b_{iv} \le \left(\overline{B}_v - \underline{B}_v\right) \sum_{j \in \hat{\mathcal{N}}_v} x_{ijv} \qquad v \in \mathcal{V}, i \in \mathcal{B}_v \tag{5.26}$$

Constraints (5.20) - (5.26) describe the bunker constraints of the model. Constraints (5.20) state that if vessel v travels directly from node i to pickup node j, then, the bunker load on board just after completing service at node i less the bunker consumption of sailing from node i to node j less the bunker consumed while loading at node j must be equal to the bunker load on board just after completing service at node j. Constraints (5.21) specify the same balance for the delivery nodes j, using the similar indexing scheme as before. Constraints (5.22) specify the bunker load balance for bunker nodes j at which there is no cargo handling. There is, however, bunker loading at node j which is added to the bunker load balance. Constraints (5.23) describe the bunker load balance during travel to the destination node as there is no bunkering nor bunker consumption in port. The quality signs in (5.20) - (5.23) ensure that no bunker is lost during travel. Constraints (5.24) specify the bunker level on board vessel v at the beginning of the planning period. Constraints (5.25) state the bunker load on board vessel v just after completing its service at node i must be greater than or equal to the minimum bunker level of the vessel plus the bunker consumed during sailing and cargo handling in node j and less than or equal to the total bunker capacity of the vessel. Constraints (5.26) ensure that the bunker purchased by vessel v at bunker node i is less than or equal to the difference between the minimum and maximum bunker load capacity. The summations in (5.25) and (5.26) are again strengthening the formulation of the model, enforcing only one of the x_{ijv} variables involved in the summation to be non-zero.

Arc Flow Constraints

$$r_{vk} = \sum_{i \in \hat{\mathcal{N}}_v} A_{ik} x_{id(v)v} \qquad k \in \mathcal{K}, v \in \mathcal{V} \tag{5.27}$$

Constraints (5.27) assign the correct values to the r_{vk} variables such that r_{vk} equals 1 if vessel v's last visited node is located in region k, and 0 otherwise. The collection of r_{vk} variables, $\boldsymbol{r}$, is used to calculate the second-stage recourse cost.

Arc Flow Binary and Non-Negativity Constraints

$$x_{ijv} \in \{0, 1\} \qquad v \in \mathcal{V}, (i, j) \in \hat{\mathcal{A}}_v \tag{5.28}$$
$$y_i \in \{0, 1\} \qquad i \in \mathcal{N}^O \tag{5.29}$$
$$z_i \in \{0, 1\} \qquad i \in \mathcal{N}^C \tag{5.30}$$
$$t_{iv} \in \mathbb{R}^+ \qquad v \in \mathcal{V}, i \in \hat{\mathcal{N}}_v \tag{5.31}$$
$$q_{iv} \in \mathbb{R}^+ \qquad v \in \mathcal{V}, i \in \mathcal{N}_v^P \tag{5.32}$$
$$b_{iv} \in \mathbb{R}^+ \qquad v \in \mathcal{V}, i \in \mathcal{B} \tag{5.33}$$
$$l_{iv}^C \in \mathbb{R}^+ \qquad v \in \mathcal{V}, i \in \hat{\mathcal{N}}_v \tag{5.34}$$
$$l_{iv}^B \in \mathbb{R}^+ \qquad v \in \mathcal{V}, i \in \hat{\mathcal{N}}_v \tag{5.35}$$
$$r_{vk} \in \{0, 1\} \qquad k \in \mathcal{K}, v \in \mathcal{V} \tag{5.36}$$

The x_{ijv} variables in Constraints (5.28) are binary as they specify whether a vessel v travels directly from node i to node j, or not. The y_i variables in Constraints (5.29) are binary as they indicate whether a spot cargo is serviced, or not. The z_i variables in Constraints (5.30) are binary as they state whether a spot ship is used to transport a contracted cargo or not. The remaining variables specified by Constraints (5.31) - (5.35) can take fractional values and are required to be non-negative. Constraints (5.31) state that the time variables t_{iv} can never take on values from before the planning period begins. Constraints (5.32) denote that a vessel v may never transport negative cargo from node i. Constraints (5.33) specify that a vessel v may never purchase a negative amount of bunker at bunker node i. Constraints (5.34) ensure that a vessel v may never have a negative cargo load on board. Constraints (5.35) state that there may never be a negative amount of bunker on board vessel v at node i. Finally, Constraints (5.36) confine the r_{vk} variables to be binary as they are designed to equal 1 if vessel v's last visited node is located in region k, and 0 otherwise.

5.3.2 Arc Flow Second-Stage Formulation

$$\mathcal{Q}(\boldsymbol{r}) = \min \quad \sum_{s\in\mathcal{S}} P_s \left(\sum_{v\in\mathcal{V}} \sum_{k\in\mathcal{K}} C^B_{d(v)k} x^B_{d(v)vks} \right) \tag{5.37}$$

s.t.

$$\sum_{k\in\mathcal{K}} x^B_{d(v)vks} \le 1 \qquad v\in\mathcal{V}, s\in\mathcal{S} \tag{5.38}$$

$$R^K_{ks} = \sum_{v\in\mathcal{V}} \left(r_{vk} + x^B_{d(v)vks} - \sum_{k'\in\mathcal{K}\setminus\{k\}} A_{d(v)k} x^B_{d(v)vk's} \right) \qquad k\in\mathcal{K}, s\in\mathcal{S} \tag{5.39}$$

$$x^B_{d(v)vks} \in \{0,1\} \qquad v\in\mathcal{V}, k\in\mathcal{K}, s\in\mathcal{S} \tag{5.40}$$

Objective (5.37) and Constraints (5.38) - (5.40) represent the stochastic second-stage recourse cost problem denoted by $\mathcal{Q}(\boldsymbol{r})$. Objective (5.37) minimizes the expected cost of repositioning vessels from their last visited node to the region they are to be allocated in aggregated over each scenario $s \in \mathcal{S}$. Constraints (5.38) state that a vessel v may only reposition to at most one region in each scenario s. Constraints (5.39) ensure that the number of vessels in each region k in each scenario s equals the predetermined parameter, R^K_{ks}. The first term in the variable expression counts up the number of vessels ending up in each region k in each scenario s as a consequence of the first-stage solution. The second term adds the number of vessels repositioning into region k in scenario s. The final term counts up the number of vessels repositioning out from region k in scenario s which is subtracted from the two previous terms. Thus, the variable expression in Constraints (5.39) correctly count up the number of vessels ending up in each region r in each scenario s. Finally, Constraints (5.40) confine the $x_{d(v)vks}$ variables to their binary domain.

5.4 Linearizations

In the presented arc flow model, there are nonlinearities in the balance constraints for time, cargo, and bunker. In addition, there are nonlinearities in Constraints (5.18) handling the discharge load capacities of the vessels, as well as in Constraints (5.25), controlling the bunker load capacities of the vessels. This section provides methods to linearize these constraints if the model is to be solved using a commercial MILP solver. In particular, Section 5.4.1 presents linearized balance constraints, while Section 5.4.2 linearizes the capacity constraints.

5.4.1 Linearized Balance Constraints

Big-M notation is introduced to linearize the balance Constraints (5.8) - (5.10), (5.13) - (5.15), and (5.20) - (5.23). As such, the multiplication of x_{ijv} variables with the balance equation can be eliminated. A large Big-M constant multiplied by $(1 - x_{ijv})$ is added or subtracted from the LHS depending on the type of inequality sign. To maintain a tight formulation, it is preferable to assign the constant with the lowest possible value that does not remove optional feasible solutions to the problem. For each constraint, the value of the Big-M should be chosen such that it makes the search space smaller by cutting away infeasible or non-optional feasible solutions.

$$t_{iv} + T^Q_{iv} q_{iv} + T^S_{ijv} - t_{jv} - M^1_{ijv}(1 - x_{ijv}) \le 0 \qquad v\in\mathcal{V}, (i,j)\in\hat{\mathcal{A}}_v \; | i\in\mathcal{N}^P_v \tag{5.41}$$

$$t_{iv} + T^Q_{iv} q_{i-N,v} + T^S_{ijv} - t_{jv} - M^2_{ijv}(1 - x_{ijv}) \le 0 \qquad v\in\mathcal{V}, (i,j)\in\hat{\mathcal{A}}_v \; | i\in\mathcal{N}^D_v \tag{5.42}$$

$$t_{iv} + T^S_{ijv} - t_{jv} - M^3_{ijv}(1 - x_{ijv}) \le 0 \qquad v\in\mathcal{V}, (i,j)\in\hat{\mathcal{A}}_v \; | i\notin\mathcal{N}^D_v \wedge i\notin\mathcal{N}^P_v \tag{5.43}$$

Constraints (5.41) - (5.43) show the resulting temporal time balance constraints after linearizing. If vessel v travels directly from node i to node j, then the Big-M term goes away. However, if vessel v does not travel directly from node i to node j, then the Big-M term is active, and must be of sufficient size for the LHS of the constraint to remain negative. The calculations of the temporal Big-M values can be found in Equations (5.44) - (5.46). In Equation (5.44), the zero inside the maximum operator ensure that the Big-M term does not contribute with a positive value if the remaining of the LHS in Constraints (5.41) is negative. The other term inside the maximum term gives the smallest value guaranteed to be greater in magnitude than the rest of the LHS in Constraint (5.41). The values are defined as the latest time vessel v may begin its service at pickup node i plus the maximum time it takes to load at node i plus the sailing time directly from node i to node j less the minimum time at which the vessel may begin service at node j. The minimum operator ensures a tighter formulation by picking the minimum of the vessel capacity or the highest possible value that can be transported. Equations (5.45) give similar values for delivery ports with a similar indexing scheme as before. Finally, Equations (5.46) specify the Big-M value for the nodes that are neither pickup or delivery nodes, where cargo handling is nonexistent.

$$M^1_{ijv} = \max\left\{0, \overline{T}_{iv} + T^Q_{iv}\min\{K_v, \overline{Q}_{iv}\} + T^S_{ijv} - \underline{T}_{jv}\right\} \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | i \in \mathcal{N}^P_v \tag{5.44}$$

$$M^2_{ijv} = \max\left\{0, \overline{T}_{iv} + T^Q_{iv}\min\{K_v, \overline{Q}_{i-N,v}\} + T^S_{ijv} - \underline{T}_{jv}\right\} \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | i \in \mathcal{N}^D_v \tag{5.45}$$

$$M^3_{ijv} = \max\left\{0, \overline{T}_{iv} + T^S_{ijv} - \underline{T}_{jv}\right\} \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | i \notin \mathcal{N}^D_v \wedge i \notin \mathcal{N}^P_v \tag{5.46}$$

Constraints (5.47) - (5.52) give the resulting cargo balance constraints after linearizing. Due to the equality sign in the origin Constraints (5.13) - (5.15), each constraint has been split into two inequalities. In the first, the Big-M term is subtracted from the LHS and used with a $\leq$ sign. In the other, the Big-M term is added to the LHS and used with a $\geq$ sign. The value of the Big-M constant is the maximum capacity, K_v, of vessel v.

$$l^C_{iv} + q_{jv} - l^C_{jv} - K_v(1 - x_{ijv}) \leq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^P_v \tag{5.47}$$

$$l^C_{iv} + q_{jv} - l^C_{jv} + K_v(1 - x_{ijv}) \geq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^P_v \tag{5.48}$$

$$l^C_{iv} - q_{jv} - l^C_{N+j,v} - K_v(1 - x_{i,N+j,v}) \leq 0 \qquad v \in \mathcal{V}, (i,N+j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^P_v \tag{5.49}$$

$$l^C_{iv} - q_{jv} - l^C_{N+j,v} + K_v(1 - x_{i,N+j,v}) \geq 0 \qquad v \in \mathcal{V}, (i,N+j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^P_v \tag{5.50}$$

$$l^C_{iv} - l^C_{jv} - K_v(1 - x_{ijv}) \leq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{B}_v \tag{5.51}$$

$$l^C_{iv} - l^C_{jv} + K_v(1 - x_{ijv}) \geq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{B}_v \tag{5.52}$$

Constraints (5.53) - (5.60) show the resulting bunker balance constraints after linearizing the original Constraints (5.20) - (5.23). The results can be explained in a similar manner as the linearization of the cargo balance constraints. The difference lies in using the maximum bunker capacity $\overline{B}_v$ of vessel v as the Big-M constant value.

$$l^B_{iv} - B^S_{ijv} - B^P_{jv}T^Q_{jv}q_{jv} - l^B_{jv} - \overline{B}_v(1 - x_{ijv}) \leq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^P_v \tag{5.53}$$

$$l^B_{iv} - B^S_{ijv} - B^P_{jv}T^Q_{jv}q_{jv} - l^B_{jv} + \overline{B}_v(1 - x_{ijv}) \geq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^P_v \tag{5.54}$$

$$l^B_{iv} - B^S_{ijv} - B^P_{iv}T^Q_{jv}q_{j-Nv} - l^B_{jv} - \overline{B}_v(1 - x_{ijv}) \leq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^D_v \tag{5.55}$$

$$l^B_{iv} - B^S_{ijv} - B^P_{iv}T^Q_{jv}q_{j-Nv} - l^B_{jv} + \overline{B}_v(1 - x_{ijv}) \geq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{N}^D_v \tag{5.56}$$

$$l^B_{iv} - B^S_{ijv} + b_{jv} - l^B_{jv} - \overline{B}_v(1 - x_{ijv}) \leq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{B}_v \tag{5.57}$$

$$l^B_{iv} - B^S_{ijv} + b_{jv} - l^B_{jv} + \overline{B}_v(1 - x_{ijv}) \geq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \mathcal{B}_v \tag{5.58}$$

$$l^B_{iv} - B^S_{ijv} - l^B_{jv} - \overline{B}_v(1 - x_{ijv}) \leq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \{d(v)\} \tag{5.59}$$

$$l^B_{iv} - B^S_{ijv} - l^B_{jv} + \overline{B}_v(1 - x_{ijv}) \geq 0 \qquad v \in \mathcal{V}, (i,j) \in \hat{\mathcal{A}}_v \; | j \in \{d(v)\} \tag{5.60}$$

5.4.2 Linearized Capacity Constraints

In the presented arc flow model, there are further nonlinearities in load capacity Constraints (5.18) and (5.25) that must be resolved to use a commercial MILP solver.

For Constraints (5.18) the nonlinearity is simply resolved by moving the q_{iv} term outside of the summation. Although this slightly weakens the original constraint, it is an efficient way of handling the nonlinearity. Constraints (5.61) give the resulting linearized constraints.

$$0 \leq l^C_{N+i,v} \leq \sum_{j \in \hat{\mathcal{N}}_v} K_v x_{N+i,jv} - q_{iv} \qquad v \in \mathcal{V}, i \in \mathcal{N}^P_v \tag{5.61}$$

In Constraints (5.25), the q_{jv} term inside the summation on the LHS is the cause of the nonlinearity. It arises due to the bunker consumed in port being dependent on the time it takes to load or discharge a load of cargo. To resolve the nonlinearity, the bunker consumed in port is rounded up to the minimum of the capacity of the ship or the largest amount of cargo that may be transported from node j. This design choice provides some additional robustness regarding the bunker level on board a vessel as it increases the lower bound of the bunker load variables. Constraint (5.62) give the linearized formulation of the original Constraints (5.25).

$$\sum_{j \in \hat{\mathcal{N}}_v} \left(\underline{B}_v + B^S_{ijv} + B^P_{jv}T^Q_{jv} \min\left\{K_v, \overline{Q}_j\right\}\right) x_{ijv} \leq l^B_{iv} \leq \overline{B}_v \sum_{j \in \hat{\mathcal{N}}_v} x_{ijv} \qquad v \in \mathcal{V}, i \in \hat{\mathcal{N}}_v \tag{5.62}$$

Chapter 6

Path Flow Solution Method

The linearized version of the arc flow model presented in Chapter 5 may be implemented and solved using commercial Mixed-Integer Linear Problem (MILP) solvers. However, due to the structure of the model, it becomes insolvable as the test instance size increases. The intractability was shown in Omholt-Jensen (2022), who solved a deterministic version of the problem which did not consider vessel repositioning decisions. Instead, this thesis implements a decomposition approach motivated by Brønmo et al. (2007b) and Omholt-Jensen (2022), incorporating a priori column generation.

By inspection of the mathematical model presented in Section 5.3, it is clear that only Constraints (5.2) and (5.3) are fleet-specific. In contrast, Constraints (5.4) - (5.27) are specific to each vessel, as there is no interaction among the different vessels for these constraints. Furthermore, each vessel's objective function splits into separate terms. As such, the structure of the problem lends itself to decomposition and column generation, in which the complex and vessel-specific constraints concerning the routing and scheduling can be handled separately in a subproblem for each vessel. The fleet-specific constraints can thus be placed in a path flow formulation with a reduced number of constraints but with a large number of columns.

This chapter elaborates on how a Dantzig-Wolfe decomposition can be applied to the arc flow model and how a priori column generation can be used with the decomposition approach to generate optimal solutions for larger problem instances. In particular, Section 6.1 presents the required notation of the problem. Section 6.2 presents the mathematical formulation of the decomposed stochastic two-stage problem resulting in a path flow formulation. Finally, Section 6.3 provides the algorithm for generating columns in the path flow problem a priori.

6.1 Path Flow Notation

The stochastic two-stage path flow formulation of the TSRSPBO incorporates the fleet-specific first-stage Constraints (5.2) and (5.3). However, the variables must be expressed by path flow variables corresponding to the optimized feasible routes for each vessel v. Let $\mathscr{N}_v$ be the set of feasible sequence of node-visits for vessel v. In this setting, feasible means the sequence of nodes adheres to Constraints (5.4) - (5.27). For each feasible sequence of nodes $\mathfrak{n} \in \mathscr{N}_v$, there is an optimal route r in which the optimal amount of cargo is transported, and the optimal of bunker is purchased to maximize the vessel-specific objective function. Let $\mathcal{R}_v$ denote the set of optimized feasible node sequences for vessel v, indexed by r. The optimized node sequences in $\mathcal{R}_v$ for vessel v are referred to as routes. Further, let the sets $\mathcal{V}$, $\mathcal{N}$, $\hat{\mathcal{N}_v}$, $\mathcal{N}^C$, $\mathcal{N}^O$, $\mathcal{K}$, and $\mathcal{S}$ be defined as in the original arc flow model presented in Chapter 5, denoting the set of vessels, cargo-related nodes, set of nodes vessel v may visit, contracted cargoes, optional spot cargoes, regions, and scenarios, respectively.

For each route $r \in \mathcal{R}_v$ for vessel v there is a vessel-specific profit denoted R_{rv}. As in the arc flow model, C_i^S denotes the cost of servicing the cargo at pickup node i with a spot ship. A new parameter, A_{irv}, is introduced and set equal to 1 if vessel v carries cargo i in the optimized route r, and 0 otherwise. Let B_{irv} equal 1 if node i is the destination node in route r of vessel v, and 0 otherwise. Let D_{ivk} be equal to 1 if node i is in region k for vessel v, and 0 otherwise. As in the arc flow model, let R^K_{ks} denote the number of vessels to be allocated in each region, r, in each scenario, s. The C^B_{ivk} parameter represents the repositioning cost from node i for vessel v to region k. Finally, let P_s be defined as before, denoting the probability of scenario s taking place.

The variable x_{rv} defines a binary variable equal to 1 if vessel v is chosen to sail route r, and 0 otherwise. Variables $y_i \in \mathcal{N}^O$ and $z_i \in \mathcal{N}^C$ are defined as before, denoting whether or not a spot cargo is serviced and whether or not a spot ship is used to service a Contract of Affreightment (CoA) cargo, respectively. Variable x^E_{iv} is defined to equal 1 if vessel v's destination is node i, and 0 otherwise. Finally, let the second-stage stochastic recourse variable x^B_{ivks} be equal to 1 if vessel v repositions from node i to region k in scenario s, and 0 otherwise.

Tables 6.1 - 6.3 summarize the notation used in the path flow formulation.

Set Notation	Set Description
$\mathcal{V}$	Set of vessels
$\mathcal{N}$	Set of all cargo-related nodes
$\hat{\mathcal{N}_v}$	Set of nodes vessel v may visit
$\mathcal{N}^C$	Set of pickup nodes for the mandatory contracted cargoes
$\mathcal{N}^O$	Set of pickup nodes for the optional spot cargoes
$\mathcal{R}_v$	Set of all routes for vessel v
$\mathcal{K}$	Set of regions
$\mathcal{S}$	Set of scenarios

Table 6.1: Path Flow Sets

Parameter Notation	Parameter Domain	Parameter Description
R_{rv}	$v \in \mathcal{V}, r \in \mathcal{R}_v$	Vessel specific profit generated by performing route r for vessel v
C^S_i	$i \in \mathcal{N}^P$	Cost of servicing the cargo at node i with a spot ship
A_{irv}	$i \in \mathcal{N}, r \in \mathcal{R}_v, v \in \mathcal{V}$	1 if vessel v carries cargo i in route r, and 0 otherwise
B_{irv}	$i \in \hat{\mathcal{N}}, r \in \mathcal{R}_v, v \in \mathcal{V}$	1 if node i is the destination node in route r of vessel v, 0 otherwise
D_{ivk}	$v \in \mathcal{V}, i \in \hat{\mathcal{N}_v}, k \in \mathcal{K}$	1 if node i is in region k for vessel v, 0 otherwise
R^K_{ks}	$k \in \mathcal{K}, s \in \mathcal{S}$	Number of vessels to be allocated in region k in scenario s
C^B_{ivk}	$i \in \hat{\mathcal{N}}, v \in \mathcal{V}, k \in \mathcal{K}$	Reposition cost from node i for vessel v to region k
P_s	$s \in \mathcal{S}$	Probability of scenario s

Table 6.2: Path Flow Parameters

Variable Notation	Variable Domain	Variable Description
x_{rv}	$r \in \mathcal{R}_v, v \in \mathcal{V}$	1 if vessel v is chosen to sail route r, 0 otherwise
y_i	$i \in \mathcal{N}^O$	1 if the optional spot cargo is transported, 0 otherwise
z_i	$i \in \mathcal{N}^P$	1 if cargo at node i is serviced by vessel from the spot market, 0 otherwise
x^E_{iv}	$i \in \hat{\mathcal{N}}, v \in \mathcal{V}$	1 if vessel v's destination is node i, 0 otherwise
x^B_{ivks}	$i \in \hat{\mathcal{N}}, v \in \mathcal{V}, k \in \mathcal{K}, s \in \mathcal{S}$	1 if vessel v repositions from node i to region k in scenario s, 0 otherwise

Table 6.3: Path Flow Variables

6.2 Path Flow Mathematical Model

This section provides the two-stage stochastic path flow formulation of the TSRSPBO studied in this thesis. The model is presented in two stages in Section 6.2.1 and Section 6.2.2, respectively.

6.2.1 Path Flow First-Stage Formulation

$$\max \quad \sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_v} R_{rv} x_{rv} - \sum_{i \in \mathcal{N}^C} C^S_i z_i - \mathcal{Q}(\boldsymbol{x^E}) \tag{6.1}$$

s.t.

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_v} A_{irv} x_{rv} + z_i = 1 \qquad i \in \mathcal{N}^C \tag{6.2}$$

$$\sum_{v \in \mathcal{V}} \sum_{r \in \mathcal{R}_v} A_{irv} x_{rv} - y_i = 0 \qquad i \in \mathcal{N}^O \tag{6.3}$$

$$\sum_{r \in \mathcal{R}_v} x_{rv} = 1 \qquad v \in \mathcal{V} \tag{6.4}$$

$$x^E_{iv} = \sum_{r \in \mathcal{R}_v} B_{irv} x_{rv} \qquad i \in \hat{\mathcal{N}}, v \in \mathcal{V} \tag{6.5}$$

$$x_{rv} \in \{0,1\} \qquad v \in \mathcal{V}, r \in \mathcal{R}_v \tag{6.6}$$

$$x^E_{iv} \in \{0,1\} \qquad i \in \hat{\mathcal{N}}, v \in \mathcal{V} \tag{6.7}$$

$$y_i \in \{0,1\} \qquad i \in \mathcal{N}^O \tag{6.8}$$

$$z_i \in \{0,1\} \qquad i \in \mathcal{N}^C \tag{6.9}$$

Objective (6.1) and Constraints (6.2) - (6.9) comprise the first-stage formulation of the TSRSPBO. The first term in Objective (6.1) sums up the vessel-specific profit generated by each of the vessels in the fleet. The second term denotes the cost of hiring a spot ship to serivce a CoA cargo and is subtracted from the vessel-specific profit generated. Finally, the second-stage $\mathcal{Q}(\boldsymbol{x^E})$ term representing the recourse cost of repositioning vessels in the fleet is subtracted from the vessel-specific profit. Here, $\boldsymbol{x^E}$ denotes the collection of x^E_{iv} variables specifying whether node i is a vessel v's destination node, or not. Constraints (6.2) state that each contracted cargo is either transported by a vessel in the fleet or by a spot ship. Constraints (6.3) specify that optional cargoes may be serviced by a vessel in the fleet. Constraints (6.4) ensure that each vessel is assigned exactly one route. Constraints (6.5) assigns the correct values to the x^E_{iv} variables such that x^E_{iv} is equal to 1 if vessel v's destination is node i, and 0 otherwise. Constraints (6.6) - (6.9) confine the binary variables to their respective domains.

6.2.2 Path Flow Second-Stage Formulation

$$\mathcal{Q}(\boldsymbol{x^E}) = \min \quad \sum_{s\in\mathcal{S}} P_s \left(\sum_{v\in\mathcal{V}} \sum_{i\in\hat{\mathcal{N}}} \sum_{k\in\mathcal{K}} C^B_{ivk} x^B_{ivks} \right) \tag{6.10}$$

s.t.

$$\sum_{k\in\mathcal{K}} x^B_{ivks} \leq x^E_{iv} \qquad v\in\mathcal{V}, i\in\hat{\mathcal{N}}_v, s\in\mathcal{S} \tag{6.11}$$

$$R^K_{ks} = \sum_{v\in\mathcal{V}} \sum_{i\in\hat{\mathcal{N}}} \left(D_{ivk} x^E_{iv} + x^B_{ivks} - \sum_{k'\in\mathcal{K}\setminus\{k\}} D_{ivk} x^B_{ivk's} \right) \qquad k\in\mathcal{K}, s\in\mathcal{S} \tag{6.12}$$

$$x^B_{ivks} \in \{0,1\} \qquad i\in\hat{\mathcal{N}}, v\in\mathcal{V}, k\in\mathcal{K}, s\in\mathcal{S} \tag{6.13}$$

Objective (6.10) and Constraints (6.11) - (6.13) define the second-stage path flow formulation of the TSRSPBO. Objective (6.10) minimizes the recourse cost which consists of the expected cost of repositioning vessels in the fleet aggregated across each scenario $s \in \mathcal{S}$. Constraints (6.11) state that a vessel may at most reposition to one region. The x^E_{iv} variables are used on the right hand side of the expression to tighten the model formulation. Constraints (6.12) ensure that the number of vessels in each region k in each scenario s equals the predetermined parameter, R^K_{ks}. The first term in the variable expression counts up the number of vessels ending up in each region k in each scenario s as a consequence of the first-stage solution. The second term adds the number of vessels repositioning into region k in scenario s. The final term counts up the number of vessels repositioning out from region k in scenario s which is subtracted from the two previous terms. Thus, the variable expression in Constraints (6.12) correctly count up the number of vessels ending up in each region r in each scenario s. Finally, Constraints (6.13) confine the x^B_{ivks} variables to their binary domain.

6.3 A Priori Column Generation with Cargo and Bunker Optimization

As input, the two-stage path flow formulation presented in Section 6.2 takes the routes generated from optimizing the feasible node sequences for a given vessel. The routes are constructed in two steps. First, all feasible node sequences for each vessel are generated. Section 6.3.1 explains the generation procedure. Second, routes are generated from each feasible node sequence. This procedure is presented in Section 6.3.2. For each feasible node sequence for each vessel, a linear programming problem is solved to determine the optimal amount of cargo to transport and the optimal amount of bunker to purchase while maximizing the vessel-specific profit function as the objective function. The linear programming problem is referred to as the Vessel Scheduling Problem (VSP). The VSP is solved for each feasible sequence of nodes for each vessel.

6.3.1 Node Sequence Generation

Feasible node sequences for each vessel are constructed from a resource constrained Depth-First-Search on the graph $\mathcal{G} = (\hat{\mathcal{N}}_v, \hat{\mathcal{A}}_v)$ between the origin node $o(v)$ and the destination node $d(v)$ for vessel v. Algorithms 1 - 3 outline the construction of feasible node sequences for a given vessel. An adjacency list stores the graph $\mathcal{G}$ such that for node i, $\mathcal{G}[i]$ returns a neighboring node j such that $(i,j) \in \hat{\mathcal{A}}_v$. Algorithm 1 explores neighboring nodes recursively until the desired sink node is reached. By starting from the origin node, $o(v)$, and setting the desired sink node to the destination node, $d(v)$, the modified Depth-First-Search explores all node sequences starting at $o(v)$ and ending at $d(v)$. To prevent cycles, an array keeps track of the visited nodes. Once $d(v)$ is reached, the current path is copied into the set of all feasible node sequences for vessel v, $\mathcal{N}_v$.

Algorithm 1: Recursive Depth-First-Search for Feasible Node Sequences

```
Data:
  $v$                                          Vessel name
  $\mathcal{G} = (\hat{\mathcal{N}}_v, \hat{\mathcal{A}}_v)$   Graph Network for vessel $v$ stored as adjecency list
  $o(v)$                                       Origin node of vessel $v$
  $d(v)$                                       Destination node of vessel $v$
  $\underline{T}_{o(v)v}$                      Time at which vessel $v$ first becomes available
  $B_v^0$                                      Initial bunker level for vessel $v$
Result:
  $\mathscr{N}_v$   Feasible node sequences for vessel $v$
Define array $visited$ of length $|\hat{\mathcal{N}}_v|$
Define arrays $path$, $time$, $bunker$, $cargo$
Define map$\langle(v, i)$ :array$\rangle$ $\mathscr{N}_v$
Define bool $feasible$
$visited[i] \leftarrow$ false for $i \ldots |\hat{\mathcal{N}}_v|$
$time[0] \leftarrow \underline{T}_{o(v),v}$
$bunker[0] \leftarrow B_v^0$
$cargo[0] \leftarrow 0$
getFeasibleNodeSequences($o(v)$, $d(v)$)

Function getFeasibleNodeSequences($i$, $d(v)$):
    $visited[i] \leftarrow$ true
    $path$.pushBack($i$)
    if $i == d(v)$ then
        $\mathscr{N}_v[(v, i)] \leftarrow$ copy($path$)
    end
    else
        for $j \in \mathcal{G}[i]$ do
            $feasible \leftarrow$ checkFeasibility($i, j, path, time, bunker, cargo$) (Algorithm 2)
            if $feasible ==$ true $\wedge visited[j] ==$ false then
                $time$, $bunker$, $cargo \leftarrow$ updateResources($time, bunker, cargo$) (Algorithm 3)
                getFeasibleNodeSequences($j, d(v)$)
            end
        end
    end
    $path$.pop()
    $time$.pop()
    $bunker$.pop()
    $cargo$.pop()
    $visited[i] \leftarrow$ false
    return
end
```

If no neighboring nodes lead to the destination node, Algorithm 1 backtracks by removing the last visited node from the path. As only feasible node sequences are of interest, Algorithm 1 keeps track of the vessel's state of time, bunker load, and cargo load. These are considered the vessel's resources. Before considering a move to a neighboring node j of node i, Algorithm 1 checks whether such a move would break any vessel-specific Constraints (5.4) - (5.27). To check any violations, Algorithm 1 makes a function call to Algorithm 2. If any constraints are broken, Algorithm 1 does not perform the move and proceeds to check the next neighboring node of i. If none of the constraints were broken, Algorithm 1 updates the vessel's resource states by making a function call to Algorithm 3 and moves to node j by performing the search recursively. Algorithm 1 shows the outline of the modified Depth-First-Search. Algorithm 2 explains how constraint violations are checked, while Algorithm 3 describes how a vessel's resource states are updated. Vessel-specific sets and parameters are available to Algorithms 1 - 3 as class members since the algorithms were implemented as class methods.

```
Algorithm 2: Check Node Sequence Feasibility
Data:
  $v$              Vessel name
  $N$              Number of cargoes
  $\mathcal{N}^D_v$    Set of delivery nodes for vessel $v$
  $\mathcal{N}^P_v$    Set of pickup nodes for vessel $v$
  $\mathcal{B}_v$      Set of bunker nodes for vessel $v$
  $K_v$            Maximum cargo capacity of vessel $v$
  $o(v)$           Origin node of vessel $v$
  $d(v)$           Destination node of vessel $v$
  $P^B_i$          Bunker price at bunker node $i$
  $\overline{T}_{iv}$   Maximum time at which vessel $v$ may begin service of node $i$
  $\underline{T}_{iv}$  Minimum time at which vessel $v$ may begin service of node $i$
  $\underline{Q}_i$     Minimum amount of cargo to transport from pickup node $i$
  $\overline{Q}_i$      Maximum amount of cargo to transport from pickup node $i$
  $T^Q_{iv}$        Time it takes of load or discharge one unit of cargo at cargo node $i$ for vessel $v$
  $T^S_{ijv}$       Time it takes to traverse arc $(i, j)$ for vessel $v$
  $B^S_{ijv}$       Bunker consumed traversing arc $(i, j)$ for vessel $v$
  $B^P_{iv}$        Bunker consumed at node $i$ for vessel $v$
  $\underline{B}_v$     Minimum bunker limit for vessel $v$
Result:
  feasible    true if feasible node sequence, false if not
Function checkFeasibility($i, j, path, time, bunker, cargo$):
    feasible ← true
                                                  /* Precedence Constraints */
    if $j \in \mathcal{N}^D_v \wedge j - N \notin path$ then feasible ← false
    if $j == d(v)$ then
        for $p \in path$ do
            if $p \in \mathcal{N}^P_v \wedge p + N \notin path$ then feasible ← false
        end
    end
                                                  /* Time Balance Constraints */
    if max(time.back(), $\underline{T}_{iv}$) $+ T^S_{ijv} \geq \overline{T}_{iv}$ then feasible ← false
    if $i \in \mathcal{N}^P_v \wedge$ max(time.back(), $\underline{T}_{iv}$) $+ T^Q_{iv}\underline{Q}_i + T^S_{ijv} \geq \overline{T}_{iv}$ then feasible ← false
    if $i \in \mathcal{N}^D_v \wedge$ max(time.back(), $\underline{T}_{iv}$) $+ T^Q_{iv}\underline{Q}_{i-N} + T^S_{ijv} \geq \overline{T}_{iv}$ then feasible ← false

                                                  /* Bunker Balance Constraints */
    if bunker.back() - $B^S_{ijv} \leq \underline{B}_v$ then feasible ← false
    if $j \in \mathcal{N}^P_v \wedge$ bunker.back() - $B^S_{ijv} - B^P_{jv}T^Q_{jv}$min$(K_v, \overline{Q}_j) \leq \underline{B}_v$ then feasible ← false
    if $j \in \mathcal{N}^D_v \wedge$ bunker.back() - $B^S_{ijv} - B^P_{jv}T^Q_{jv}$min$(K_v, \overline{Q}_{j-N}) \leq \underline{B}_v$ then feasible ← false

                                                  /* Cargo Balance Constraints */
    if $j \in \mathcal{N}^P_v \wedge$ cargo.back()+$\underline{Q}_j \geq K_v$ then feasible ← false
    if cargo.back() $\geq K_v$ then feasible ← false

    return feasible
end
```

One significant limitation of Algorithm 1 is related to the prevention of cycles. As the recursive Depth-First-Search algorithm does not allow for cycles during the construction of feasible node sequences, it might exclude solutions that can be found by the arc flow model presented in Section 5.3. Algorithm 1 might exclude optimal solutions because, in theory, it could be optimal for a vessel to visit the same bunker node multiple times. In the TSRSPBO studied in this thesis, the bunker nodes are available for the entire duration of the planning horizon. As such, a one-to-one correspondence between a bunker port and a bunker node exists. A simple solution to allow for

Algorithm 3: Update Resources

Data:

v	Vessel name
N	Number of cargoes
$\mathcal{N}_v^D$	Set of delivery nodes for vessel v
$\mathcal{N}_v^P$	Set of pickup nodes for vessel v
$\mathcal{B}_v$	Set of bunker nodes for vessel v
K_v	Maximum cargo capacity of vessel v
$\underline{T}_{iv}$	Minimum time at which vessel v may begin service of node i
$\underline{Q}_i$	Minimum amount of cargo to transport from pickup node i
$\overline{Q}_i$	Maximum amount of cargo to transport from pickup node i
T_{iv}^Q	Time it takes of load or discharge one unit of cargo at cargo node i for vessel v
T_{ijv}^S	Time it takes to traverse arc (i,j) for vessel v
B_{ijv}^S	Bunker consumed traversing arc (i,j) for vessel v
B_{iv}^P	Bunker consumed at node i for vessel v
$\overline{B}_v$	Maximum bunker limit for vessel v

Result:

time	updated time `array`
bunker	updated time `array`
cargo	updated time `array`

```
Function updateResources($i$, $d(v)$):
                                                            /* Time Update */
    if $i \in \mathcal{N}_v^P$ then $time$.pushBack(max($time$.back(), $\underline{T}_{iv}$) $+ T_{iv}^Q \underline{Q}_i + T_{ijv}^S$)
    else if $i \in \mathcal{N}_v^D$ then $time$.pushBack(max($time$.back(), $\underline{T}_{iv}$) $+ T_{iv}^Q \underline{Q}_{i-N} + T_{ijv}^S$)
    else $time$.pushBack(max($time$.back(), $\underline{T}_{iv}$) $+ T_{ijv}^S$)

                                                            /* Bunker Update */
    if $j \in \mathcal{N}_v^P$ then $bunker$.pushBack($bunker$.back() $-B_{ijv}^S - B_{jv}^P T_{jv}^Q$min($K_v, \overline{Q}_j$))
    else if $j \in \mathcal{N}_v^D$ then $bunker$.pushBack($bunker$.back() $-B_{ijv}^S - B_{jv}^P T_{jv}^Q$min($K_v, \overline{Q}_{j-N}$))
    else if $j \in \mathcal{B}_v$ then $bunker$.pushBack($\overline{B}_v$)
    else $bunker$.pushBack($bunker$.back() $-B_{ijv}^S$)

                                                            /* Cargo Update */
    if $j \in \mathcal{N}_v^P$ then $cargo$.pushBack($cargo$.back()$+\underline{Q}_j$)
    else if $j \in \mathcal{N}_v^D$ then $cargo$.pushBack($cargo$.back()$-\underline{Q}_j$)
    else $cargo$.pushBack($cargo$.back())

    return $time$, $bunker$, $cargo$
end
```

multiple bunker visits would be to split the bunker ports into multiple bunker nodes such that the bunker port's time windows are partitioned into non-overlapping time windows of any arbitrary duration. By denoting this duration as D^B, the Depth-First-Search will allow for multiple visits to the same port as long as it is not within the duration of D^B. As this duration could be made arbitrarily small, it can be decreased until it is within a realistic size. However, the drawback of this approach is the increased number of nodes in the model and, consequently, the increased time complexity of the solving problem. As such, the implemented path flow solution method presented in this chapter does not include further discretization of bunker nodes according to their time windows. It should be noted that this is easily achievable if discrepancies are found between the optimal arc flow solutions and the Depth-First-Search approach.

In Algorithm 2, each potential move from node i to node j is checked for feasibility according to Constraints (5.4) - (5.26). In line 2, the boolean variable denoting feasibility is set to `true`. If any constraints are broken, the variable is set to `false,` denoting that Algorithm 1 should skip to the next neighbor of i. Lines 3-8 correspond to the cargo precedence constraints presented in

Constraints (5.7) and (5.11). Line 3 states that a vessel may not visit a delivery node before it has visited its corresponding pickup node. Line 6 ensures that a vessel may only complete a valid sequence of nodes after visiting all delivery nodes corresponding to the pickup nodes in its path. Note that it is unnecessary to explicitly enforce the network flow Constraints (5.4) - (5.6) as they are enforced implicitly by the Depth-First-Search's traversal procedure. Lines 9-11 make sure that the time balance equations presented in Constraints (5.8) - (5.10) and (5.12) are not violated. In lines 10 and 11, the minimum cargo quantity to be transported, $\underline{Q}_i$, is used to calculate the time spent in port conservatively. Lines 13-15 enforce the bunker balance constraints presented in Constraints (5.20) - (5.26). In lines 14 and 15, the conservative parameter of the maximum cargo to be transported, $\overline{Q}_i$, is used when calculating the bunker consumed in port. Finally, lines 17 and 18 exclude all moves violating Constraints (5.13) - (5.19). Note that the minimum cargo quantity to be transported, $\underline{Q}_i$, is used to check for feasibility in line 17.

If a move from node i to a neighboring node j is feasible with respect to Algorithm 2, a vessel's resource states are updated according to Algorithm 3. In lines 2-4, the time balance is updated depending on whether node i is a pickup node, a delivery node, or any other node. In lines 6-9, the bunker balance on board the vessel is updated depending on whether node j is a pickup node, delivery node, bunker node, or any other node. Finally, lines 11-13 update the cargo on board the vessel depending on whether node j is a pickup node, delivery node, or any other node.

Algorithm 1 generates a set of all feasible node sequences $\mathscr{N}_v$ for vessel v upon completion. Algorithm 1 is run for each vessel in the fleet. Once all feasible node sequences are found for every vessel in the fleet, the next step is to optimize the node sequences to generate optimal routes.

6.3.2 Route Generation

Given the set of all feasible node sequences, $\mathscr{N}_v$, for vessel v, the Vessel Scheduling Problem (VSP) is solved for each feasible node sequence, $\mathfrak{n}$, for each vessel, v, to generate the optimal route, r. As the node sequence generation procedure explained in Section 6.3.1 eliminates node sequences not abiding by the network flow constraints represented by Constraints (5.4) - (5.7), they can be excluded from the VSP model formulation. Constraints (5.11) are also eliminated as these precedence constraints are ensured by the node sequence generation. Moreover, the variables x_{ijv} are no longer useful as the nodes the vessel visits are specified by the nodes for a given node sequence. Thus, the x_{ijv} variables are also excluded from the VSP formulation. The subscripts v are eliminated from the sets, parameters, and variables defined for the arc flow model presented in Section 5.3, as each model is defined for a given node sequence for a specific vessel. Otherwise, the sets, parameters, and variables carry the same meaning as defined in Section 5.2. Finally, to define required indexing notation, let $\mathfrak{n}$ be a given node sequence for a specific vessel. Let $\mathscr{K}$ be the indexing set of $\mathfrak{n}$ such that $\mathscr{K} = \{0, 1, \ldots, |\mathfrak{n}| - 1\}$, indexed by k. Further, let $i(k)$ be the node of $\mathfrak{n}$ at index k. By making these adjustments to the arc flow model presented in Chapter 5, the mathematical VSP formulation can be presented. The Objective Function (6.14) and Constraints (6.15) - (6.39) define the VSP problem. The problem is a linear programming problem with no integer or binary variables. As such, it is considerably easier to solve than the arc flow model presented in Chapter 5. Solving the VSP for each feasible node sequence for each vessel yields the sets of routes $\mathcal{R}_v$ for each vessel v. The objective function value associated with each optimal schedule s is the vessel-specific optimal profit R_{rv} for route r used in the path flow formulation.

Objective Function

$$\max \sum_{k \in \mathscr{K} | i(k) \in \mathcal{N}^P} R_{i(k)} q_{i(k)} - \sum_{k \in \mathscr{K}} C_{i(k),i(k+1)} - \sum_{k \in \mathscr{K} | i(k) \in \mathcal{B}} P_{i(k)} b_{i(k)} + P(l_d^B - B^0) \tag{6.14}$$

The Objective Function (6.14) maximizes the cargo quantity dependent revenue generated less the cost of traversing the route, the cost of purchasing bunker, and the difference in bunker quantity on board the vessel priced at the average bunker fuel price. The fixed sailing cost is included as the objective function value of the optimal solution becomes the R_{rv} parameter used in the path flow formulation.

Temporal Constraints

$$t_{i(k)} + T^Q_{i(k)} q_{i(k)} + T^S_{i(k),i(k+1)} - t_{i(k+1)} \leq 0 \qquad k \in \mathscr{K} | i(k) \in \mathcal{N}^P \tag{6.15}$$

$$t_{i(k)} + T^Q_{i(k)} q_{i(k)-N} + T^S_{i(k),i(k+1)} - t_{i(k+1)} \leq 0 \qquad k \in \mathscr{K} | i(k) \in \mathcal{N}^D \tag{6.16}$$

$$t_{i(k)} + T^S_{i(k),i(k+1)} - t_{i(k+1)} \leq 0 \qquad k \in \mathscr{K} | i(k) \notin \mathcal{N}^D \wedge i(k) \notin \mathcal{N}^P \tag{6.17}$$

$$\underline{T}_{i(k)} \leq t_{i(k)} \leq \overline{T}_{i(k)} \qquad k \in \mathscr{K} \tag{6.18}$$

The temporal constraints defined by (6.15) - (6.18) ensure the time is updated correctly at the different nodes in the route and that the time limits of the nodes in the route are adhered to. They correspond to Constraints (5.8) - (5.12) in the original model.

Cargo Constraints

$$l^C_{i(k)} + q_{i(k+1)} - l^C_{i(k+1)} = 0 \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{N}^P \tag{6.19}$$

$$l^C_{i(k)} - q_{i(k+1)-N} - l^C_{i(k+1)} = 0 \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{N}^D \tag{6.20}$$

$$l^C_{i(k)} - l^C_{i(k+1)} = 0 \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{B} \tag{6.21}$$

$$l^C_o = 0 \tag{6.22}$$

$$q_{i(k)} \leq l^C_{i(k)} \leq K \qquad k \in \mathscr{K} | i(k) \in \mathcal{N}^P \tag{6.23}$$

$$0 \leq l^C_{i(k)+N} \leq K - q_{i(k)} \qquad k \in \mathscr{K} | i(k) \in \mathcal{N}^P \tag{6.24}$$

$$\underline{Q}_{i(k)} \leq q_{i(k)} \leq \overline{Q}_{i(k)} \qquad k \in \mathscr{K} | i(k) \in \mathcal{N}^P \tag{6.25}$$

The Constraints (6.19) - (6.21) correspond to Constraints (5.13) - (5.15) and make sure that the cargo on board the vessel is updated correctly. Constraint (6.22) sets the initial cargo amount on board a vessel to zero and corresponds to Constraints (5.16). Constraints (6.23) and (6.24) ensure the cargo quantity transported remains within vessels' cargo capacity limits at pickup nodes and delivery nodes, respectively. They correspond to Constraints (5.17) and (5.18). Finally, Constraints (6.25) correspond to Constraints (5.19) and make certain that the amount of cargo loaded is within the MoLOO limits.

Bunker Constraints

$$l^B_{i(k)} - B^S_{i(k),i(k+1)} - B^P_{i(k+1)} T^Q_{i(k+1)} q_{i(k+1)} - l^B_{i(k+1)} = 0 \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{N}^P \tag{6.26}$$

$$l^B_{i(k)} - B^S_{i(k),i(k+1)} - B^P_{i(k+1)} T^Q_{i(k+1)} q_{i(k+1)-N} - l^B_{i(k+1)} = 0 \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{N}^D \tag{6.27}$$

$$l^B_{i(k)} - B^S_{i(k),i(k+1)} + b_{i(k+1)} - l^B_{i(k+1)} = 0 \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{B} \tag{6.28}$$

$$l^B_{i(k)} - B^S_{i(k),i(k+1)} - l^B_{i(k+1)} = 0 \qquad k \in \mathscr{K} | i(k+1) = d \tag{6.29}$$

$$l^B_o = B^0 \tag{6.30}$$

$$\underline{B} + B^S_{i(k),i(k+1)} + B^P_{i(k+1)} T^Q_{i(k+1)} q_{i(k+1)} \leq l^B_{i(k)} \leq \overline{B} \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{N}^P \tag{6.31}$$

$$\underline{B} + B^S_{i(k),i(k+1)} + B^P_{i(k+1)} T^Q_{i(k+1)} q_{i(k+1)-N} \leq l^B_{i(k)} \leq \overline{B} \qquad k \in \mathscr{K} | i(k+1) \in \mathcal{N}^D \tag{6.32}$$

$$\underline{B} + B^S_{i(k),i(k+1)} \leq l^B_{i(k)} \leq \overline{B} \qquad k \in \mathscr{K} | i(k+1) \notin \mathcal{N}^D \wedge i(k+1) \notin \mathcal{N}^P \tag{6.33}$$

$$0 \leq b_{i(k)} \leq \overline{B} - \underline{B} \qquad k \in \mathscr{K} | i(k) \in \mathcal{B} \tag{6.34}$$

Constraints (6.26) - (6.29) correspond to Constraints (5.20) - (5.23), and make sure that the bunker balance is updated correctly. Constraint (6.30) specifies the initial bunker level and corresponds to Constraints (5.24). Constraints (6.31) - (6.33) states that the amount of bunker carried by the vessel is within its bunker capacity limits and correspond to Constraints (5.25). Finally, Constraints (6.34) correspond to Constraints (5.26) and specify that the amount of bunker purchased is within the bunker capacity limits.

Non-Negativity Constraints

$$t_{i(k)} \in \mathbb{R}^+ \qquad k \in \mathscr{K} \tag{6.35}$$

$$q_{i(k)} \in \mathbb{R}^+ \qquad k \in \mathscr{K} | i(k) \in \mathcal{N}^P \tag{6.36}$$

$$b_{i(k)} \in \mathbb{R}^+ \qquad k \in \mathscr{K} | i(k) \in \mathcal{B} \tag{6.37}$$

$$l^C_{i(k)} \in \mathbb{R}^+ \qquad k \in \mathscr{K} \tag{6.38}$$

$$l^B_{i(k)} \in \mathbb{R}^+ \qquad k \in \mathscr{K} \tag{6.39}$$

Constraints (6.35) - (6.39) confine their respective variables to their domain. They correspond to Constraints (5.31) - (5.35).

Chapter 7

Iterative Matheuristic for Path Flow Models

The study by Omholt-Jensen (2022) concludes that the path flow formulation solution method, in combination with a priori route generation, can solve test instances with 30 cargoes, ten vessels, and ten bunker ports for a problem similar to this thesis within 30 minutes to optimality. This thesis extends the problem to a two-stage stochastic programming model by also considering which regions vessels end up in at the end of the planning period. Since Western Bulk's long-term fleet consists of 20-30 vessels, a solution method that can handle larger test instances than solved in Omholt-Jensen (2022) is needed. To address this challenge, this thesis proposes a heuristic approach that can deliver good solutions within a reasonable time. The proposed approach is an iterative matheuristic based on the Adaptive Large Neighborhood Search (ALNS) framework proposed by Ropke and Pisinger (2006). However, it should be emphasized that the ALNS is incorporated mainly as a means of generating columns for the path flow problem formulated in Section 6.2. The path flow problem is referred to as the Vessel Combination Problem (VCP). Once the VCP is solved, the best obtained solution is fed back into the ALNS procedure for further exploration of the search space. The iterative matheuristic combining ALNS and VCP is motivated by Ulsrud et al. (2022) and Archetti et al. (2012), that both leverage iterative heuristic mechanisms to find columns to a path flow problem. While the solution approach proposed by Ulsrud et al. (2022) relies heavily on the ALNS framework, Archetti et al. (2012) leverage an iterative matheuristic that solves a path flow model at regular intervals. The iterative matheuristic proposed in this thesis combines the solution approaches put forth by Ulsrud et al. (2022) and Archetti et al. (2012).

As the problem studied in this thesis incorporates optimal bunker decisions, it is no longer a traditional Ship Routing and Scheduling (SRS) problem which is mainly focused on the routing and scheduling of cargo-related nodes. As a consequence, the iterative matheuristic solution approach proposed by this thesis relies more heavily on the combination of local search and solving the VCP at regular intervals than on the ALNS framework. The ALNS is incorporated to explore the search space and provide the VCP with an increasing number of vessel-specific routes to select from. In this thesis, the iterative matheuristic solution approach is referred to as ALNS for the VCP (ALNS-VCP).

Section 7.1 presents a high-level overview of the proposed ALNS-VCP matheuristic. Section 7.2 explains how an initial solution is constructed. Section 7.3 outlines the large neighborhood search process, a core component of the ALNS procedure. Section 7.4 presents local search extensions incorporated in the ALNS-VCP. Section 7.5 provide details of how the ALNS-VCP matheuristic is adapted to handle the stochastic two-stage problem formulation presented in Section 6.2.2. Section 7.6 explains how the VCP is solved at regular intervals based on the columns encountered by the ALNS procedure. Section 7.7 outlines the acceptance criterion leveraged in each iteration of the ALNS-VCP to preserve the diversification of the search. Section 7.8 describes how noise is introduced to the ALNS-VCP to diversify the search. Finally, Section 7.9 presents the adaptive selection process of the destroy and repair operators applied during the ALNS procedure.

7.1 Overview of the ALNS

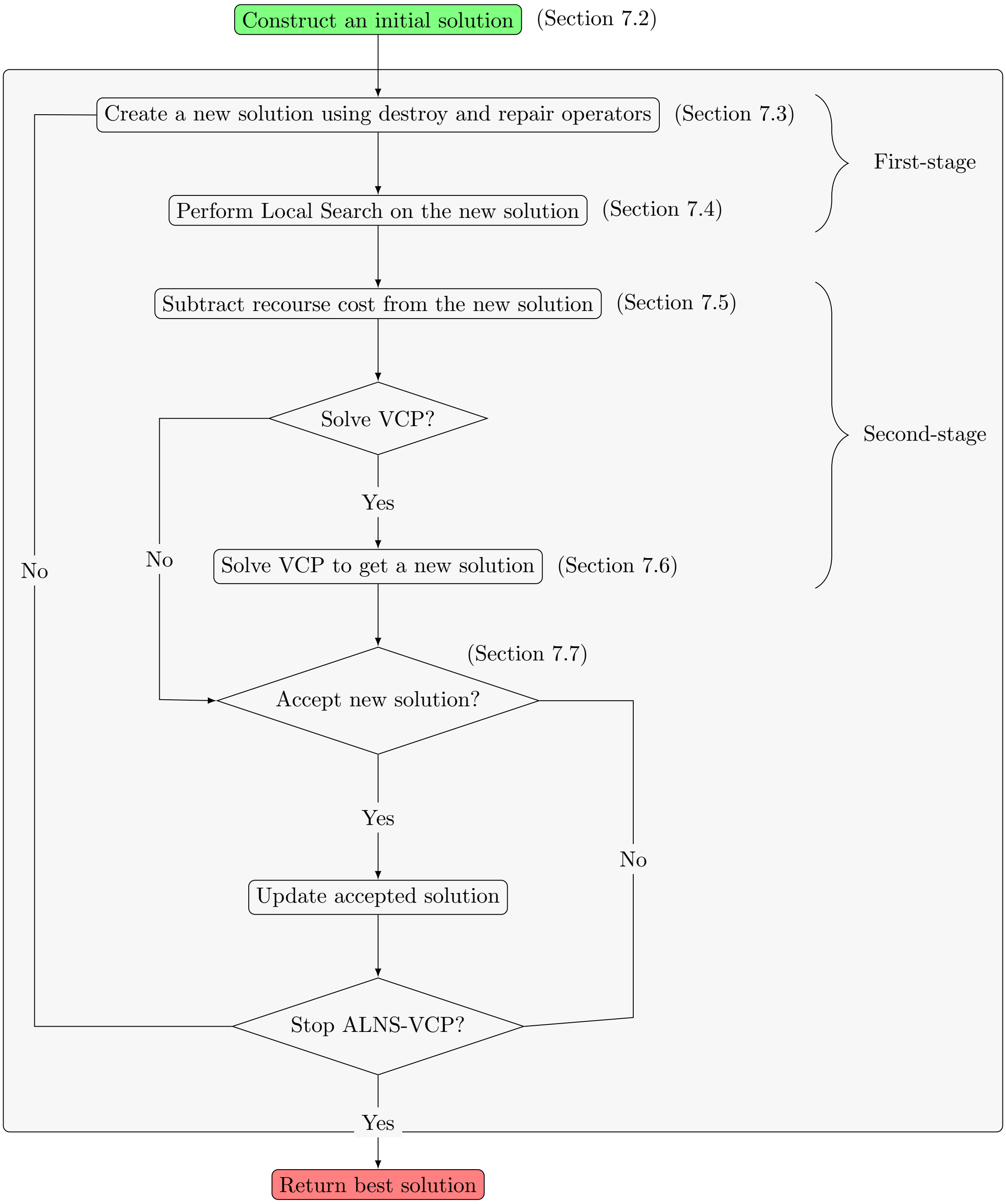


Figure 7.1: Flow chart representing the different components of the iterative ALNS-VCP matheuristic.

Figure 7.1 provides a high-level overview of the implemented ALNS-VCP matheuristic. The algorithm begins by constructing an initial solution. Next, a combination of destroy and repair operators are applied in sequence to generate a candidate solution $x^{candidate}$. To explore other feasible solutions in the local neighborhood of $x^{candidate}$, local search is applied to the $x^{candidate}$ solution. The sequence of destroy and repair operators, as well as the local search operators, aims to find solutions to the first-stage path flow problem that is described in Section 6.2.1.

Algorithm 4: Adaptive Large Neighborhood Search for the Vessel Combination Problem (ALNS-VCP)

```
Data:
  $I^{ALNS-VCP}$   Number of ALNS-VCP iterations
  $I^{S}$          Number of iterations in a segment
  $I^{VCP}$        Number of iterations after which the VCP is solved
Result:
  $x^{best}$   Best found solution
set current accepted solution $x^{accepted}$ by constructing an initial feasible solution
set the best-found solution $x^{best} \leftarrow x^{accepted}$
set the current segment $m \leftarrow 1$
set adaptive weights $w_{dm}$ equal to 1 for each destroy operator $d$
set adaptive weights $w_{rm}$ equal to 1 for each repair operator $r$
calculate selection probabilities $P_{dm}, P_{rm}$ based on $w_{dm}, w_{rm}$, respectively

for iteration = 1 to $I^{ALNS-VCP}$ do
    set current candidate solution $x^{candidate} \leftarrow x^{accepted}$
    select a destroy operator using selection probabilities $P_{dm}$
    select a repair operator using selection probabilities $P_{rm}$
    apply the destroy and repair operators on $x^{candidate}$
    apply local search operators on $x^{candidate}$
    subtract the second-stage cost from the candidate solution's objective function

    if $I^{VCP}$ iterations have passed since the VCP was solved then
        improve the candidate solution $x^{candidate}$ by solving the VCP
    end

    if $x^{candidate}$ is accepted according to simulated annealing criterion then
        $x^{accepted} \leftarrow x^{candidate}$
    end

    if $f(x^{accepted}) > f(x^{best})$ then
        apply local search operators on $x^{accepted}$
        improve the accepted solution $x^{accepted}$ by solving the VCP
        $x^{best} \leftarrow x^{accepted}$
    end

    update scores $\pi_d, \pi_r$ for the repair and destroy operators, respectively

    if $I^{S}$ iterations have passed since last weight update then
        update weights $w_{d,m+1}$ and $w_{r,m+1}$ to be used in segment $m+1$ based on scores $\pi_d, \pi_r$,
          respectively
        calculate new selection probabilities $P_{d,m+1}$ and $P_{r,m+1}$ based on $w_{d,m+1}$ and $w_{r,m+1}$,
          respectively
        update the simulated annealing temperature, $T$
        update current segment, $m \leftarrow m+1$
    end
end
```

After the local search operators are applied to the $x^{candidate}$ solution, the expected second-stage recourse cost is deducted from the profit of $x^{candidate}$ discovered during the first-stage solution search. The cost can be calculated by solving the second-stage path flow problem formulated in Section 6.2.2. Additionally, the Vessel Combination Problem (VCP) is solved in every I^{VCP} iteration. The VCP is solved using all the routes encountered during the ALNS procedure, and its solution replaces $x^{candidate}$. The VCP differs from the overall path flow solution method presented in Section 6 because it is only solved for a subset of all feasible routes.

Once the first-stage and the second-stage phases of the ALNS-VCP matheuristic in Figure 7.1 have been completed, the candidate solution $x^{candidate}$ is checked to determine whether it should be accepted as the accepted solution $x^{accepted}$. The acceptance criterion of the implemented ALNS-VCP follows a simulated annealing approach. Therefore, candidate solutions may be accepted even though they are worse than the best-known solution. If a candidate solution is accepted, $x^{accepted} \leftarrow x^{candidate}$, and otherwise $x^{candidate}$ is discarded. Finally, the ALNS-VCP stop criterion determines whether the matheuristic should continue or return the best-found solution.

Algorithm 4 provides a more detailed pseudocode of the implemented matheuristic. In particular, it introduces the process of selecting destroy and repair operators in each iteration. Operators are selected based on probabilities and weights that are updated depending on the success of each operator in finding promising solutions.

7.2 Constructing an Initial Feasible Solution

The ALNS-VCP matheuristic starts by constructing an initial solution for the provided test instance. Let $\mathcal{U}$ denote the set of cargoes not assigned to any vessel in the fleet. The construction heuristic begins by setting $\mathcal{U} \leftarrow \mathcal{N}^P$, the set of pickup nodes. Additionally, the initial routes for each vessel are set to only include the origin and destination node, $o(v)$ and $d(v)$, respectively. At each iteration, a cargo, represented by the pickup and delivery node-pair $(i, i+N)$, is inserted into a vessel's route at the position that increases the fleet-specific profit by the most. Subsequently, i is removed from $\mathcal{U}$. As such, the construction heuristic implements a greedy insertion criterion. the construction heuristic terminates when either $\mathcal{U} = \emptyset$ or there are no feasible cargo insertions.

Table 7.1 shows the resulting vessel routes for running the construction heuristic on a test instance with two vessels and five cargoes. For illustrative purposes, bunker nodes are ignored. Note that the origin and destination nodes are represented by 0 and 11, respectively.

Iteration	0	1	2	3	4
$\mathcal{N}^P$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$
$\mathcal{N}^D$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$
$\mathcal{U}$	$\{1,2,3,4,5\}$	$\{1,2,4,5\}$	$\{2,4,5\}$	$\{2,5\}$	$\{2\}$
v_1	$[0,11]$	$[0,3,8,11]$	$[0,3,8,11]$	$[0,4,3,9,8,11]$	$[0,4,3,9,8,11]$
v_2	$[0,11]$	$[0,11]$	$[0,1,6,11]$	$[0,1,6,11]$	$[0,1,6,5,10,11]$

Table 7.1: Numerical example of construction heuristic

In iteration 3, cargo (4, 9) is inserted into vessel v_1's route such that the two cargoes (3, 8) and (4, 9) are transported as parcels, meaning that they are transported simultaneously. Further, the construction heuristic terminates after iteration 4, as the cargo (2, 7) cannot be inserted into any vessel's route.

The construction heuristic must check the potential route's profitability when deciding what cargo to insert in which vessel's route in what position. Each potential route must also be checked for feasibility with regard to the time balance, cargo balance, and bunker balance constraints specified in Constraints (5.8) - (5.26). Additionally, the bunker node that contributes most to a vessel's route profit must be inserted at its best position. Algorithm 5 presents a pseudocode for calculating the profit of a potential cargo insertion. The algorithm is an essential part of the ALNS-VCP and is utilized by the construction heuristic and the destroy and repair operators.

In lines 4 and 13 of Algorithm 5, a potential route is checked for feasibility with regard to Constraints (5.8) - (5.26). In lines 5 and 14, the LP model presented in Section 6.3.2 is solved for a potential route to generate the optimal vessel-specific profit. Note that Algorithm 5 inserts at most one bunker node. However, as the input *route* may contain bunker nodes, the resulting *bestRoute* may contain more than one bunker node.

Table 7.2 illustrates the results of running the construction heuristic on the same test instance as presented in Table 7.1 when including bunker nodes. The set $\mathcal{B} = \{11, 12\}$ denotes bunker nodes. Consequently, the destination node $d(v)$ takes on the value of 13.

Iteration	0	1	2	3	4
$\mathcal{N}^P$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$
$\mathcal{N}^D$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$	$\{6,7,8,9,10\}$
$\mathcal{B}$	$\{11,12\}$	$\{11,12\}$	$\{11,12\}$	$\{11,12\}$	$\{11,12\}$
$\mathcal{U}$	$\{1,2,3,4,5\}$	$\{1,2,4,5\}$	$\{2,4,5\}$	$\{2,5\}$	$\{2\}$
v_1	$[0,13]$	$[0,3,11,8,13]$	$[0,3,11,8,13]$	$[0,4,3,9,11,8,13]$	$[0,4,3,9,11,8,13]$
v_2	$[0,13]$	$[0,13]$	$[0,1,6,12,13]$	$[0,1,6,12,13]$	$[0,1,6,12,5,10,13]$

Table 7.2: Numerical example of construction heuristic with bunker nodes

```
Algorithm 5: Bunker Insertion Search
Data:
  route    Candidate route for vessel v
  B        Set of bunker nodes
Result:
  bestRoute    Best route for vessel v
  bestProfit   Best found profit for vessel v
Function bunkerInsertionSearch(route, $\mathcal{B}$):
    Define a set allRoutes sorted by increasing profit

    if route is feasible with respect to Equations (5.8) - (5.26) then
        Get profit for route by solving the LP model in Section 6.3.2
        route.profit ← profit
        allRoutes.insert(route)
    end
    for $b \in \mathcal{B}$ do
        for i = 1 to route.size() -1 do
            bunkerRoute ← route
            bunkerRoute.insert(b, i)
            if bunkerRoute is feasible with respect to Equations (5.8) - (5.26) then
                Get profit for bunkerRoute by solving the LP model in Section 6.3.2
                bunkerRoute.profit ← profit
                allRoutes.insert(bunkerRoute)
            end
        end
    end
    Define bestRoute ← allRoutes.last()
    Define bestProfit ← bestRoute.profit
    return bestRoute, bestProfit
end
```

Once the cargo assignment phase of the construction heuristic terminates, the solution's profit is calculated. The vessel-specific profits are summed to obtain the fleet-specific profit. For each pickup node, i, such that $i \in \mathcal{N}^C \cap \mathcal{U}$, the cost of hiring a spot vessel to service a CoA cargo is subtracted from the fleet-specific profit to obtain the solution profit. Finally, as explained in Section 7.5, the second-stage recourse cost of repositioning vessels into their respective regions is deducted from the solution profit.

7.3 Large Neighborhood Search

In each iteration of the ALNS-VCP matheuristic, the current solution is modified by choosing and applying, first, a destroy operator and, subsequently, a repair operator. The process of destroying and repairing a solution is referred to as the large neighborhood search component of the ALNS procedure. In Section 7.3.1, the implemented destroy operators are presented. Section 7.3.2 provides details about the repair operators implemented in this thesis.

7.3.1 Destroy Operators

In total, three destroy operators were implemented for the ALNS procedure. These draw inspiration from those presented in Ropke and Pisinger (2006) but are adapted to the specific problem studied in this thesis. The three implemented destroy operators are named *random removal*, *Shaw removal*, and *worst removal*.

Before applying any of the destroy operators, how much of the current solution to destroy must be determined. Let q denote the number of cargoes to remove from the current solution, and let $\mathcal{Q}$ denote the set of cargoes to remove. Further, let $\mathcal{T}$ denote the set of cargoes serviced in the current solution such that $\mathcal{T} = \mathcal{N}^P \setminus \mathcal{U}$. Finally, q is decided by drawing an integer number from the discrete uniform distribution, $Unif[1, \alpha|\mathcal{T}|]$, where $\alpha \in \mathbb{R} \mid 1 < \alpha|\mathcal{T}| \wedge \alpha < 1$ is a constant.

Random Removal
Given q as the number of cargoes to remove from the current solution, *random removal* randomly samples q cargoes from $\mathcal{T}$ using a uniform distribution. The sampled cargoes are assigned to the set $\mathcal{Q}$.

Shaw Removal
The *Shaw removal* destroy operator was first proposed by Shaw (1998) and further adapted to Pickup and Delivery Problems with Time Windows (PDPTWs) by Ropke and Pisinger (2006). The idea of the operator is to remove cargoes related to each other. This approach can be advantageous since, when a repair operator reinserts the removed cargoes into the vessel routes, the cargoes' similarities provide additional insertion options. Hopefully, some of these possibilities will result in an improved solution.

Let $R(i, j)$ denote the relatedness of two cargoes, i and j. The relatedness function $R(i, j)$ is defined as

$$R(i,j) = \frac{\big(d^P(i,j) + d^D(i,j)\big) - D^{min}}{D^{max} - D^{min}} + \frac{\big(|\underline{T}_i - \underline{T}_j| + |\overline{T}_i - \overline{T}_j|\big) - T^{min}}{T^{max} - T^{min}} \tag{7.1}$$

where $d^P(i, j)$ and $d^D(i, j)$ denote the shortest sailable geospherical distance between the pickup nodes and delivery nodes of cargoes i and j, respectively. $\underline{T}_i$ and $\underline{T}_j$ denote the pickup time for cargoes i and j averaged over the fleet of vessels. Similarly, $\overline{T}_i$ and $\overline{T}_j$ denote the delivery time for cargoes i and j. The distance terms and the time terms are normalized to be in a range between zero and one by leveraging the minimum and maximum values across all cargoes. As such, define $D^{min} = \min\{d^P(i, j) + d^D(i, j)\}$ across all cargoes and $D^{max} = \max\{d^P(i, j) + d^D(i, j)\}$ across all cargoes. T^{max} and T^{min} are defined similarly for the time windows.

The pseudocode for the Shaw removal is outlined in Algorithm 6. A parameter $p^{\geq}1$ is introduced to allow for some randomness when selecting related cargoes. A higher value of p leads to selecting the most related cargo more often. Conversely, a value of $p = 1$ leads to selecting cargoes at random without using the relatedness measure at all. As output, Algorithm 6 returns the set $\mathcal{Q}$, which represents the cargoes that are to be removed from the current solution.

Algorithm 6: Shaw Removal as presented in Ropke and Pisinger (2006)

```
Data:
  $q$    Number of cargoes to remove from current solution
  $p$    Randomness parameter
  $\mathcal{T}$    Set of serviced cargoes in current solution
Result:
  $\mathcal{Q}$    Set of cargoes to remove from current solution

Function shawRemoval($q$, $p$, $\mathcal{T}$):
    Define set of cargoes, $\mathcal{Q} \leftarrow \{\}$
    while $|\mathcal{Q}| < q$ do
        Define $r \leftarrow$ a randomly selected cargo from $\mathcal{T}$
        Define array $L \leftarrow$ the cargoes not in $\mathcal{Q}$
        Sort $L$ such that $i < j \Rightarrow R(r, L[i]) < R(r, L[j])$
        Choose a random number $y$ from the interval [0, 1)
        Define $index \leftarrow \lfloor y^p \cdot |L| \rfloor$          /* Higher $p$ leads to lower $index$ */
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{L[index]\}$
    end
    return $\mathcal{Q}$
end
```

Worst Removal

The *worst removal* destroy operator aims to remove cargoes that contribute the least to the current solution's profit. The motivation for removing cargoes of low contribution is that they might represent cargoes that are placed in the wrong position. As such, subsequently applied repair operators might replace the removed cargo by a more profitable alternative. Let $cont(i, s)$ denote the contribution of cargo $i \in \mathcal{T}$ in the current solution, s. The contribution function is defined as $cont(i, s) = f(s) - f_{-i}(s)$, where $f(s)$ denotes the profit of the current solution and $f_{-i}(s)$ denotes the profit of the current solution when cargo i is removed.

Algorithm 7 presents pseudocode for the worst removal procedure. The p parameter is again introduced to allow for some randomness when selecting the worst cargo. On completion, Algorithm 7 returns the set $\mathcal{Q}$, representing the cargoes to be removed from the current solution.

Algorithm 7: Worst removal

```
Data:
  $s$    Current solution
  $q$    Number of cargoes to remove from current solution
  $p$    Randomness parameter
  $\mathcal{T}$    Set of serviced cargoes in current solution
Result:
  $\mathcal{Q}$    Set of cargoes to remove from current solution

Function worstRemoval($q$, $p$, $\mathcal{T}$, $s$):
    Define set of cargoes, $\mathcal{Q} \leftarrow \{\}$
    while $|\mathcal{Q}| < q$ do
        Define array $L \leftarrow$ the cargoes in $\mathcal{T}$
        Sort $L$ by ascending $cont(i, s)$
        Choose a random number $y$ from the interval [0, 1)
        Define $index \leftarrow \lfloor y^p \cdot |L| \rfloor$          /* Higher $p$ leads to lower $index$ */
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{L[index]\}$
    end
    return $\mathcal{Q}$
end
```

The presented removal operators select a set of cargoes to remove from the current solution. Given a vessel route in the current solution that services a cargo to be removed, removing the corresponding pickup and delivery nodes from the vessel's route is straightforward. However, the vessel route also contains bunker nodes. These complicate the removal of cargoes as visiting a bunker node might no longer be optimal after the removal of cargoes. As such, when performing the removal of cargoes, all bunker nodes in a vessel's route are also removed. This removal might result in an infeasible route as a vessel would run out of bunker if it were to service the remaining cargoes.. Infeasibility is a problem because the repair operators presented in Section 7.3.2 relies on the destroyed solution's profit when attempting to repair the destroyed solution. As such, bunker nodes are purposefully inserted back into the vessel routes in cargo to make the destroyed vessel routes feasible. The bunker insertions follow a similar procedure as presented in Algorithm 5 to insert the bunker node at the position maximizing the vessel-specific profit. Finally, the set of cargoes not assigned to any vessel in the fleet, $\mathcal{U}$, is updated to reflect the removed cargoes by assigning $\mathcal{U} \leftarrow \mathcal{U} \cup \mathcal{Q}$.

7.3.2 Repair Operators

This thesis implements a single repair operator that can be parameterized to yield multiple operator variants. The choice of repair operator was motivated by Ropke and Pisinger (2006). However, the implementation is adapted to fit the specific problem studied in this thesis. This section presents the implemented *regret-k insertion* operator and explains how it can be parameterized to yield different variants.

Regret-k insertion
The *regret-k insertion* operator attempts to repair the destroyed solution by inserting cargoes from the set of unassigned cargoes, $\mathcal{U}$, into the destroyed vessel routes. As in the construction heuristic, vessel routes are modified by inserting the corresponding pickup and delivery nodes sequentially. However, the regret-k insertion operator is not limited to appending a cargo to a current vessel's route but may insert at any position in the route. The k parameter defines the preference for how greedy the repair operator should act. By using $k = 1$, the cargoes increasing the destroyed solution's objective value by the most is inserted. Higher values of k result in insertions that try to incorporate look-ahead information that aims to avoid local optima.

Let $\Delta f(s, i, j, v)$ denote the change in the objective value of solution, s, by inserting a cargo, $i \in \mathcal{U}$, in position, j, in vessel v's route such that the resulting route is feasible. For each cargo $i \in \mathcal{U}$, let $\boldsymbol{\Delta}_i$ denote the collection of $\Delta f(s, i, j, v)$ sorted in descending cargo for cargo i. Let $\boldsymbol{\Delta}_i$ be indexed by k.

Next, define the *regret-k value* as

$$c_i^*(k) = \begin{cases} \Delta f(s,i,j,v) & \text{if } k = 1 \\ \sum_{l=1}^{k} \left(\boldsymbol{\Delta}_i[0] - \boldsymbol{\Delta}_i[l]\right) & \text{if } k > 1 \end{cases} \tag{7.2}$$

The regret-k insertion operator chooses the cargo insertion that maximizes

$$\max_{i \in \mathcal{U}} c_i^*(k). \tag{7.3}$$

In the case that $k = 1$, the operator inserts the cargo, i, in position, j, of the vessel v's route, which improves the objective value of the destroyed solution by the most. As such, it is greedy in the pure sense of the word, inserting the most profitable cargo without considering other insertions. For higher values of k, the operator calculates the regret value as defined in Equation (7.2). The regret value measures how much one would regret not inserting the kth best move. As the value of k increases, it becomes evident sooner that there are fewer options to insert a cargo. The implemented regret-k insertion operator handles regret value tiebreaks by inserting the cargo with the highest contribution to the objective value of the solution. Further, if $\exists i \in \mathcal{U}$ such that $|\boldsymbol{\Delta}_i| < k$, the regret value cannot be calculated for cargo i. In such cases, the cargo i with the smallest $|\boldsymbol{\Delta}_i|$ is inserted with tiebreaks broken the same way as for regret values. In addition to inserting a cargo $i \in \mathcal{U}$, the implemented regret-k operator inserts the bunker node at the position in the route that increases the vessel-specific profit by the most. The procedure of finding which bunker node to insert in which position follows the pseudocode outlined in Algorithm 5. As the regret-k operator inserts a cargo i, it is removed from the set of unassigned cargoes, $\mathcal{U}$. The operator terminates once $\mathcal{U} = \emptyset$ or there are no more feasible cargo insertions.

7.4 Local Search Extension

The destroy and repair operators introduced in Section 7.3 aim to explore a diverse set of locations in the solution search space. The large neighborhood search process represented by the destroy and repair operators attempts to mitigate the chance of getting stuck in local optima. However, the search for high-quality solutions can be improved by performing exhaustive searches in the local neighborhood of the solutions resulting from the large neighborhood search. The iterative ALNS-VCP matheuristic implemented in this thesis leverages three local search operators at each iteration of the algorithm. First, the ALNS-VCP applies the CROSS-Exchange operator, introduced to Vehicle Routing Problems with Time Windows (VRPTWs) by Taillard et al. (1997), to each resulting solution from the large neighborhood search. Second, a proposed Bunker-Destination operator is applied to handle the specific nature of the problem studied in this thesis.

Section 7.4.1 outlines the details of the implemented CROSS-Exchange operator. Finally, Section 7.4.2 introduces the proposed Bunker-Destination operator.

7.4.1 CROSS-Exchange Operator

The CROSS-exchange operator involves selecting two pairs of edges in two different routes, and exchanging them in a way that generates two new routes. Specifically, four nodes i, k, j, l are selected such that i and k belong to the same route, and j and l belong to a different route. Next, the edges $(i, i{+}1)$ and $(j, j{+}1)$ are swapped to produce edges $(i, j{+}1)$ and $(j, i{+}1)$. Similarly, edges $(k{-}1, k)$ and $(l{-}1, l)$ are also swapped to generate edges $(l{-}1, k)$ and $(k{-}1, l)$. Figure 7.2 illustrates the procedure which produces two new routes. The CROSS-Exchange operator essentially swaps substrings of cargoes between two vessel routes. To ensure that the resulting routes are feasible with respect to the cargo precedence constraints presented by Network Flow Constraints (5.4) - (5.7) and Time Balance Constraints (5.11), each substring is checked for feasibility before a substring exchange takes place. If a substring contains a pickup node but not its corresponding delivery not, the substring exchange is skipped. Similarly, if a substring contains a delivery node but not its corresponding pickup node, the substrings are also not exchanged.

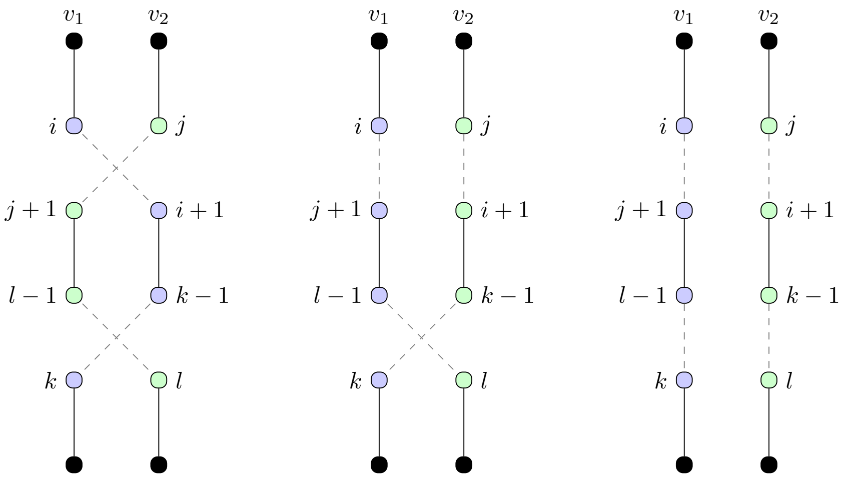


Figure 7.2: Illustration of a CROSS-Exchange substring exchange of cargoes between two vessel routes

Each route generated by the substring exchanges is further checked for feasibility with respect to Time, Cargo, and Bunker Balance Constraints (5.8) - (5.26). Finally, the associated profit is calculated by solving the LP problem formulated in Section 6.3.2. It should be noted that the implemented CROSS-Exchange operator does not change the routes in the current solution. Rather, every feasible route resulting from the substring exchanges and their respective profits are added to a pool, $pool^{ALNS}$, of encountered routes during the ALNS procedure. As explained in Section 7.1, the Vessel Combination Problem (VCP) is solved every I^{VCP} iteration with $pool^{VCP}$ as input. Therefore, routes only encountered by the CROSS-Exchange operator are never propagated to the current solution until the VCP is solved.

7.4.2 Proposed Bunker-Destination Operator

A way that vessels might mitigate the expected cost of repositioning to their respective regions would be to travel to a bunker node in the correct region after serving their assigned cargoes. To aid the ALNS-VCP matheuristic in finding such solutions, a local search operator inserting bunker nodes at the end of a vessel's route was implemented. For each bunker node, $b \in \mathcal{B}$, the Bunker-Destination local search operator first checks if a bunker node is currently at the position prior to the destination node in vessel v's route. If so, the bunker node is removed before b is inserted in its place. The resulting route is checked for feasibility with respect to Constraints (5.8) - (5.26). If feasible, the profit is calculated by solving the LP problem presented in Section 6.3.2. Finally, feasible routes and their associated profits are inserted into $pool^{ALNS}$.

Algorithm 8: Bunker-Destination Operator

```
Data:
  $V$              Set of vessels
  $\mathcal{B}$              Set of bunker nodes
  $pool^{ALNS}$    Pool of routes and profits encountered by the ALNS procedure
Result:
  $pool^{ALNS}$    Updated pool of routes encountered by the ALNS procedure

Function bunkerDestinationSearch($\mathcal{V}$, $pool^{ALNS}$):
    for $v = 0$ to $|\mathcal{V}|$ do
        for $b \in \mathcal{B}$ do
            Define $newRoute \leftarrow$ current route of vessel $v$
            if a bunker node is at the end of $newRoute$ then
                Remove the bunker node from $newRoute$
            end
            Insert $b$ prior to the destination node in $newRoute$
            if $newRoute$ is feasible with respect to Equations (5.8) - (5.26) then
                Get $profit$ for $newRoute$ by solving the LP model in Section 6.3.2
                Add $newRoute$ and $profit$ to $pool^{ALNS}$
            end
        end
    end
end
```

7.5 Second-Stage Cost

The large neighborhood search represented by the application of destroy and repair operators on the candidate solution, $x^{candidate}$, as well as the subsequent local search represented by the CROSS-Exchange and Bunker-Destination operators, comprises the first-stage phase of the implemented ALNS-VCP matheuristic. To account for the cost of repositioning vessels into the required regions, the second-stage recourse cost must be deducted from the candidate solution, $x^{candidate}$'s objective function value. This section presents the second-stage problem solved in each iteration of the ALNS-VCP matheuristic. The formulation differs slightly from the model presented in Section 6.2.2, as each vessel in the candidate solution is already assigned a single route. As such, a more compact formulation of the second-stage problem was implemented to lessen the time complexity of the ALNS-VCP matheuristic.

Let the sets $\mathcal{V}, \mathcal{N}^C, \mathcal{N}^O, \mathcal{K}, \mathcal{S}$ be defined as in Section 6.1 denoting the set of vessels, contracted CoA cargoes, optional spot cargoes, regions, and scenarios, respectively. A parameter B_{vk} is introduced which is equal to 1 if the vessel v's destination node is in region k, and 0 otherwise. Let C^B_{vk} denote the cost of repositioning vessel, v, from its destination node to region k. Let R^K_{ks} be defined as in Section 6.1, signifying the number of vessels to be allocated in each region, k, in each scenario, s. The parameter P_s denotes the probability of scenario s being realized. Finally, the binary variable x^B_{vks} equals 1 if vessel, v, repositions from its destination node to region, k, in scenario, s, and 0 otherwise. A summary of the presented notation is provided in Tables 7.3 - 7.5.

Set Notation	Set Description
$\mathcal{V}$	Set of vessels
$\mathcal{N}^C$	Set of pickup nodes for the mandatory contracted cargoes
$\mathcal{N}^O$	Set of pickup nodes for the optional spot cargoes
$\mathcal{K}$	Set of regions
$\mathcal{S}$	Set of scenarios

Table 7.3: ALNS-VCP Second-Stage Sets

Parameter Notation	Parameter Domain	Parameter Description
B_{vk}	$v \in \mathcal{V}, k \in \mathcal{K}$	1 if vessel v's destination is in region k, 0 otherwise
C^B_{vk}	$v \in \mathcal{V}, k \in \mathcal{K}$	Reposition cost from vessel v's destination node to region k
R^K_{ks}	$k \in \mathcal{K}, s \in \mathcal{S}$	Number of vessels to be allocated in region k in scenario s
P_s	$s \in \mathcal{S}$	Probability of scenario s

Table 7.4: ALNS-VCP Second-Stage Parameters

Variable Notation	Variable Domain	Variable Description
x^B_{vks}	$v \in \mathcal{V}, k \in \mathcal{K}, s \in \mathcal{S}$	1 if vessel v repositions from its destination node to region k in scenario s, 0 otherwise

Table 7.5: Path Flow Variables

$$\min \quad \sum_{s \in \mathcal{S}} P_s \left(\sum_{v \in \mathcal{V}} \sum_{k \in \mathcal{K}} C^B_{vk} x^B_{vks} \right) \tag{7.4}$$

s.t.

$$\sum_{k \in \mathcal{K}} x^B_{vks} \leq 1 \qquad v \in \mathcal{V}, s \in \mathcal{S} \tag{7.5}$$

$$R^K_{ks} = \sum_{v \in \mathcal{V}} \left(B_{vk} + x^B_{vks} - \sum_{k' \in \mathcal{K} \setminus \{k\}} B_{vk} x^B_{vk's} \right) \qquad k \in \mathcal{K}, s \in \mathcal{S} \tag{7.6}$$

$$x^B_{vks} \in \{0, 1\} \qquad v \in \mathcal{V}, k \in \mathcal{K}, s \in \mathcal{S} \tag{7.7}$$

The resulting ALNS-VCP second-stage formulation is represented by Objective (7.4) and Constraints (7.5) - (7.7). Objective (7.4) minimizes the expected recourse cost of repositioning vessels to the required regions. Constraints (7.5) ensure that vessels are only allowed to reposition to a single region. Constraints (7.6) ensure that the number of vessels in each region k in each scenario s equals the predetermined parameter, R^K_{ks}. The first term in the variable expression counts up the number of vessels ending up in each region by summing the B_{vk} parameters. The second term adds the number of vessels repositioning into region k in scenario s. The final term counts up the number of vessels repositioning out from region k in scenario s which is subtracted from the two previous terms. Finally, Constraints (6.13) confine the x^B_{vks} variables to their binary domain.

7.6 Vessel Combination Problem (VCP)

After the ALNS procedure has explored the search space for I^{VCP} iterations, the Vessel Combination Problem (VCP) is solved to identify the best combination of feasible vessel-specific routes encountered during the ALNS procedure. The resulting solution is fed back into the ALNS for further search diversification.

In the a priori path flow solution method presented in Chapter 6, all feasible node sequences for each vessel are generated as outlined in Section 6.3.1. Further, as outlined in Section 6.3.2, the respective profit is calculated by solving the LP problem formulated in Section 6.3.2. The process yields the set of routes $\mathcal{R}_v$ for each vessel, v, and the vessel-specific profit, R_{rv}, for each vessel, v, an route $r \in \mathcal{R}_v$. Let the path flow pool, $pool^{PF}$, denote the collection of sets $\mathcal{R}_v$ and respective vessel-specific profits, R_{rv}, for each vessel, v. The two-stage path flow model presented in section 6.2 is solved to generate optimal solutions by taking $pool^{PF}$ as input.

During the construction of an initial feasible solution and during the application of destroy, repair, and local search operators, a vast number of vessel routes are checked for feasibility with regards to Constraints (5.8) - (5.26). If feasible, the vessel-specific profit is calculated by solving the LP problem formulated in Section 6.3.2. The process is outlined in Algorithm 5. Rather than discarding all encountered feasible routes and their respective profits, they are kept by inserting them into a pool, $pool^{ALNS}$.

When comparing $pool^{PF}$ with $pool^{ALNS}$ it becomes evident that $|pool^{PF}| > |pool^{ALNS}|$. Nevertheless, the idea is that the ALNS procedure should be capable of searching for a diverse set of vessel-specific routes, which are stored in $pool^{ALNS}$, such that the VCP can alleviate the task of combining encountered routes into a fleet-specific combination of routes, constituting a feasible solution.

However, solving the VCP in every iteration introduces significant time complexity to the ALNS-VCP matheuristic. Furthermore, it is not certain that a significant number of new vessel routes have been encountered within a single iteration that could be recombined into a higher-quality solution. As such, the VCP is solved at regular intervals, at every I^{VCP} iteration of the ALNS-VCP matheuristic.

7.7 Acceptance Criterion

The accepted criterion implemented in the ALNS-VCP presented in this thesis leverages a simulated annealing-based approach similar to Ropke and Pisinger (2006). At the beginning of an ALNS-VCP iteration, a candidate solution, $x^{candidate}$ is generated by applying the destroy, repair, and local search operators on the currently accepted solution $x^{accepted}$. Rather than restricting the search operators to only be applied to $x^{accepted}$ if it is the best solution found during the search, a simulated annealing criterion lets the search operators be applied to a variety of different solutions. Such an approach should hopefully mitigate the chance for the ALNS-VCP matheuristic to get trapped in local optima.

Given a candidate solution, $x^{candidate}$, with profit, $f(x^{candidate})$ and accepted solution, $x^{accepted}$ with profit, $f(x^{accepted})$, the ALNS-VCP matheuristic assigns $x^{candidate}$ to $x^{accepted}$ with probability, $p = 1$, if $f(x^{candidate}) > f(x^{accepted})$. Otherwise, if $f(x^{candidate}) \leq f(x^{accepted})$, the ALNS-VCP matheuristic assigns $x^{candidate}$ to $x^{accepted}$ with probability $p = e^{-(f(x^{candidate})-f(x^{accepted}))/T}$ where T is referred to as the *temperature*. The temperature is initially set to T^{start} and is decreased by assigning $T \leftarrow T \cdot c$ every iteration where $0 < c < 1$ is the *cooling rate*. As such, the simulated annealing criterion allows for a diverse set of solutions to be accepted during the early stages of the ALNS-VCP matheuristic. However, in the later iterations of the ALNS-VCP matheuristic, when the temperature is low, $x^{candidate}$ with $f(x^{candidate}) \leq f(x^{accepted})$ is only accepted with a small probability. As such, as the search progresses, the acceptance criterion promotes the intensification of the ALNS-VCP matheuristic.

7.8 Applying Noise in the Insertion Operators

Ropke and Pisinger (2006) describe their repair operators as being quite myopic. To combat this behavior, they introduce a noise term when calculating the contribution of a potential reparation. As their studied problem is a cost minimization problem for a Vehicle Routing Problem with Time Windows (VRPTW), their noise application procedure was tailored to the specific Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) studied in this thesis. Let $\Delta f(s, i, j, v)$ denote the change in the objective value of solution, s, by inserting a cargo, $i \in \mathcal{U}$, in position, j, in vessel v's route such that the resulting route is feasible. When calculating $\Delta f(s, i, j, v)$, one must decide whether to account for a cargo being a Contract of Affreightment (CoA) cargo or not. For example, if one decides to account for CoA cargoes, one would calculate $\Delta f(s, i, j, v)$ by adding the cost of hiring a spot ship to service the cargo. After all, by servicing a CoA cargo by a vessel in the fleet, no spot ship cost would be incurred, and vice versa. Always accounting for the spot ship cost was found to make the repair operators presented in Section 7.3.2 quite myopic. The repair operators tend to always favor CoA cargoes. As such, noise was applied to the $\Delta f(s, i, j, v)$ by randomly choosing whether to account for the spot ship cost or not.

7.9 Adaptive Selection of Destroy and Repair Operators

In each iteration of the ALNS-VCP matheuristic, a single destroy operator is selected from the set of implemented destroy operators, and a single repair operator is selected from the set of implemented repair operator variants. The selection is part of the large neighborhood search of the ALNS procedure. To this end, a roulette wheel selection is leveraged for the destroy and repair operators, respectively. The weights associated with each operator are adjusted based on the operator's success rate in finding promising solutions. The repeated update of these weights comprises the adaptivity of the ALNS procedure as proposed by Ropke and Pisinger (2006). To facilitate the repeated update of weights, the search process is divided into segments, denoted as m, with each segment consisting of a certain number of iterations, referred to as I^S. At the start of a new segment, the weights are updated. For each destroy operator, denoted as d, the weights are adjusted using

$$w_{dm} = (1 - \widetilde{r})w_{d,m-1} + \widetilde{r}\frac{\pi_d}{\theta_d}$$

.

Here, the new weight value is a weighted average of the previous weights $w_{d,m-1}$, and a reaction term comprising the tunable reaction parameter $\widetilde{r}$, the operator score π_d, and the number of times θ_d that the destroy operator d has been selected in segment $m - 1$. At the start of each new segment, the destroy operator score π_d is reset to zero and can increase during the iterations of the current segment in three different ways:

1. If a candidate solution becomes the best-found solution, the destroy operator score π_d is increased by σ_1.
2. If a candidate solution has a higher objective value than the accepted solution and has not been accepted before, the destroy operator score π_d is increased by σ_2.
3. If a candidate solution has not been accepted before and is worse than the current solution but is still accepted by the simulated annealing criterion, the destroy operator score π_d is increased by σ_3.

For each repair operator, denoted as r, the update of weights w_{rm} works in the same manner by instead using terms $w_{r,m-1}, \pi_r$, and θ_r.

By denoting I_d as the set of all destroy operators, the probability of selecting destroy operator d

in the roulette wheel selection process of segment m is determined by

$$P(d,m) = \frac{w_{d,m}}{\sum_{\hat{d} \in I_d} w_{\hat{d},m}}$$

.

Similarly, for I_r, the set of repair operators, the probability of selecting repair operator r is

$$P(r,m) = \frac{w_{r,m}}{\sum_{\hat{r} \in I_r} w_{\hat{r},m}}$$

.

Chapter 8

Test Instances and Implementation

This chapter elaborates on retrieving real-life data from industry partners, constructing test instances, and generating vessel networks in the presented models. In particular, Section 8.1 explains how the data required by the Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) studied in this thesis was retrieved. Section 8.2 showcases an example of a constructed test instance and presents the generated test instances. Finally, Section 8.3 explains a few techniques utilized when implementing the solution methods presented in Chapters 6 and 7.

8.1 Input Data

This section explains how real-life cargo and voyage data from the industry partner, Western Bulk, was compiled into suitable test instances. In addition to Western Bulk's operational data, routing and port information were provided by Maritime Optima, a maritime data analytics company. Section 8.1.1 explains how input was retrieved from these industry partners. The remaining input data needed for the model was either gathered from open sources, constructed from open sources, estimated from available data, or randomly sampled from reasonable uniform intervals. Section 8.1.2 explains this process.

8.1.1 Industry Partner Input Data

Table 8.1 shows a small sample of the data provided by Western Bulk. In total, the data consisted of some 9,200 voyages performed in the time period 2015-2022. Due to commercial reasons, only examples from 2015 are shown in this thesis. From the provided data, voyages performed by vessels in the subsegments Supramax and Ultramax were extracted. These subsegments were chosen as they are usually on longer Time Charter (TC) contracts, a part of Western Bulk's operation they are looking to improve. Parameter values such as freight rates, R_i, and More or Less in Owner's Option (MoLOO) flexibility limits $\left[\underline{Q}_i, \overline{Q}_i\right]$ of transported cargo can be difficult to come by, so working with real-life data provided by Western Bulk makes constructing realistic test instances an easier task.

Date	2015-10	2015-10	2015-12	2015-12	2015-12
Vessel Name	PRABHU GOPAL	ASTORIA	WESTERN HEROYA	SPAR LYRA	MAINE DREAM
DWT	56025	63353	61297	53565	58105
Ship Contract Type	Period TC	Trip TC	Index	Period TC	Period TC
Voyage Type	coa	coa	spot	spot	spot
Route Type	Pacific_RV	Cont_FarEast	Atlantic_RV	Cont_FarEast	Cont_FarEast
Cargo	Petcoke	Steel Slabs	WET FLYASH	Steel Rebars	Wheat
Cargo Quantity	10980	57721	49909	32177	51435
Freight Rate	41.0	23.0	12.8	24.75	22.25
Voyage Commence Port	zhangzhou	itaguai	lorient	icdas	sauda
Pickup Port	zhenjiang	itaguai	brunsbuttel	icdas	hamburg
Delivery Port	sohar	los angeles	jacksonville	chimbote	djibouti

Table 8.1: Example data of Western Bulk voyages

Western Bulk also provided its definition of regions associated with each port. Figure 8.1 shows a map of ports and their associated regions in the Indian Ocean.

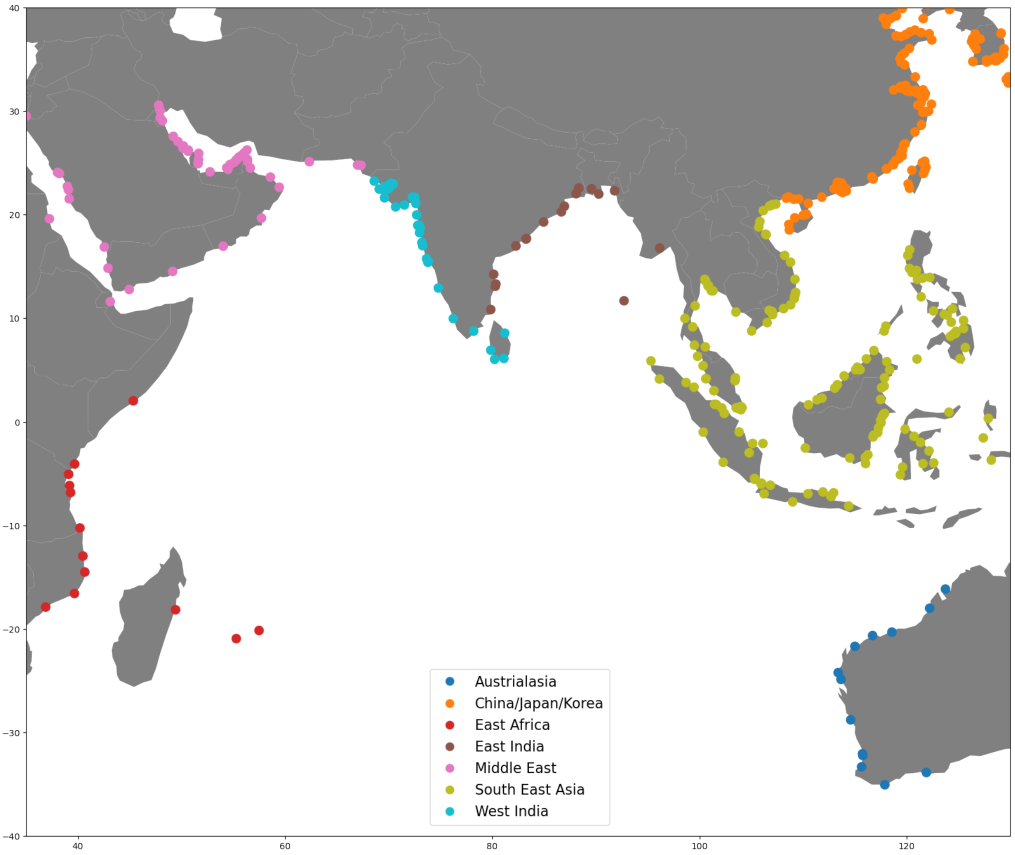


Figure 8.1: Port regions provided by Western Bulk

The model presented in Chapter 5 further relies on distances between ports, their locations, and the price of purchasing one unit of bunker at bunker ports. Geo-locational port data and distance data were provided by MaritimeOptima (2022). Bunker price data was extracted through Maritime Optima's integration with BunkerEx (2022). Figure 8.2 shows simplified trajectories corresponding to the shortest path of voyages performed by Western Bulk in the time period 2015-2022. Figure 8.3 shows bunker prices of different fuel grades for some of the most popular bunker ports (Husby, 2022). This thesis only considers a single type of fuel, Very Low Sulfur Fuel Oil (VLSFO), as it is seeing increasing adoption in the shipping industry. Due to the environmental regulations of IMO (2020), shipping companies are increasingly purchasing VLSFO rather than cheaper and more polluting Intermediate Fuel Oil with Maximum Viscosity of 380 Centistokes (IFO380).

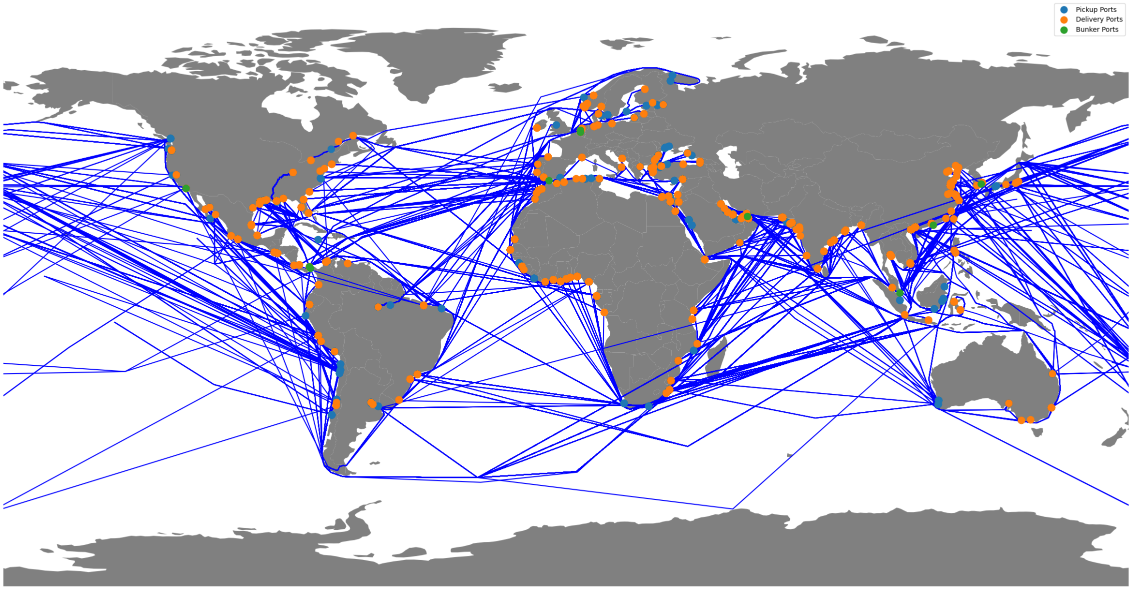
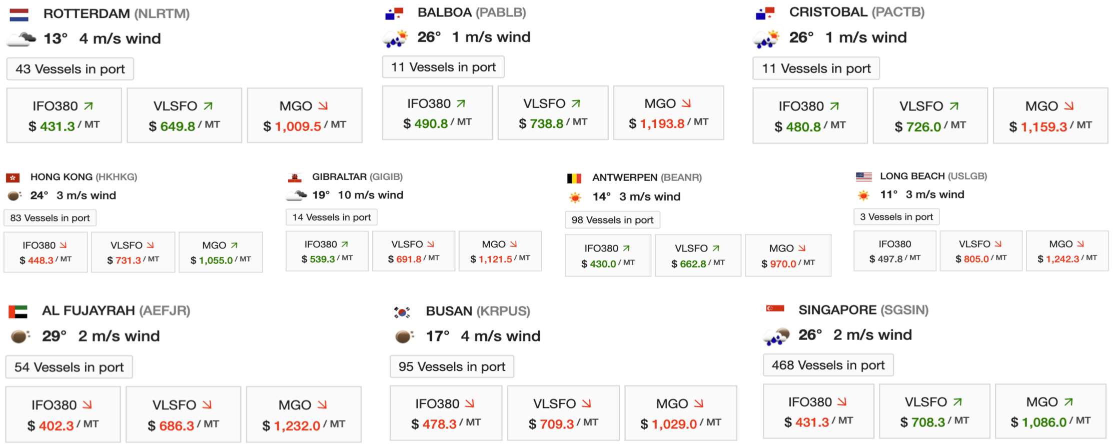


Figure 8.2: Routes corresponding to the shortest path between ports for voyages performed by Western Bulk (MaritimeOptima, 2022)



ROTTERDAM (NLRTM)
13° 4 m/s wind
43 Vessels in port
IFO380 ↗ $ 431.3 / MT
VLSFO ↗ $ 649.8 / MT
MGO ↘ $ 1,009.5 / MT

BALBOA (PABLB)
26° 1 m/s wind
11 Vessels in port
IFO380 ↗ $ 490.8 / MT
VLSFO ↗ $ 738.8 / MT
MGO ↘ $ 1,193.8 / MT

CRISTOBAL (PACTB)
26° 1 m/s wind
11 Vessels in port
IFO380 ↗ $ 480.8 / MT
VLSFO ↗ $ 726.0 / MT
MGO ↘ $ 1,159.3 / MT

HONG KONG (HKHKG)
24° 3 m/s wind
83 Vessels in port
IFO380 ↘ $ 448.3 / MT
VLSFO ↘ $ 731.3 / MT
MGO ↗ $ 1,055.0 / MT

GIBRALTAR (GIGIB)
19° 10 m/s wind
14 Vessels in port
IFO380 ↗ $ 539.3 / MT
VLSFO ↘ $ 691.8 / MT
MGO ↘ $ 1,121.5 / MT

ANTWERPEN (BEANR)
14° 3 m/s wind
98 Vessels in port
IFO380 ↗ $ 430.0 / MT
VLSFO ↗ $ 662.8 / MT
MGO ↘ $ 970.0 / MT

LONG BEACH (USLGB)
11° 3 m/s wind
3 Vessels in port
IFO380 $ 497.8 / MT
VLSFO ↘ $ 805.0 / MT
MGO ↘ $ 1,242.3 / MT

AL FUJAYRAH (AEFJR)
29° 2 m/s wind
54 Vessels in port
IFO380 ↘ $ 402.3 / MT
VLSFO ↘ $ 686.3 / MT
MGO ↘ $ 1,232.0 / MT

BUSAN (KRPUS)
17° 4 m/s wind
95 Vessels in port
IFO380 ↘ $ 478.3 / MT
VLSFO ↘ $ 709.3 / MT
MGO ↘ $ 1,029.0 / MT

SINGAPORE (SGSIN)
26° 2 m/s wind
468 Vessels in port
IFO380 ↘ $ 431.3 / MT
VLSFO ↗ $ 708.3 / MT
MGO ↗ $ 1,086.0 / MT

Figure 8.3: Bunker prices (retrieved November 11th, 2022) for the top ten most popular ports (BunkerEx, 2022)

Along with data provided by Western Bulk and Maritime Optima, a few open sources were used to compile additional necessary data. Canal costs were compiled from the Suez Canal Toll Calculator from Wilhelmsen (2022), and spot ship rates were gathered from Fearnleys (2022), who publish weekly Supramax spot ship rates for several popular trading routes.

8.1.2 Constructed Input Data

Although input data provided by the industry partners has been a tremendous help when compiling real-life test instances, several parameter data were challenging to attain. Some of these were gathered by talking with Western Bulk and Maritime Optima industry experts to define a sensible range to which the parameters were confined. These ranges were thus uniformly randomly sampled during test instance generation to imitate fluctuations and ascertain the model's flexibility. Some industry expert-generated parameter values include port costs, vessel speeds, bunker consumption, time in ports, and the number of vessels to allocate in each region.

A few parameters were estimated based on available data from Western Bulk. These include the $\left[\underline{Q}_i, \overline{Q}_i\right]$ parameters describing the MoLOO flexibility limits of transported cargo, the port of which a spot ship is available, the origin ports, $o(v)$, of the vessels, and the $C^B_{d(v)k}$ parameters denoting the cost of repositioning from vessel v's destination node to region k. The MoLOO limits were estimated by creating a ± 10% interval from the actual amount of transported cargo. The limits were rounded down and up to the closest integer of 1,000. Further, the port at which a spot ship would become available to transport a CoA cargo was estimated by finding the most frequent "voyage commence port" (see Table 8.1) for a given pickup port. A voyage commence port is the name Western Bulk uses for ports of which a vessel becomes available and thus is ready to take on new cargo. Cases where the voyage commence port and the pickup port were identical were omitted as it is improbable that a spot ship would be available in the same port as the cargo. The origin ports were found similarly by looking at the most frequent delivery ports of the data provided by Western Bulk. Although parceling occurs, most vessels will be empty at their delivery port, making them available to take on new cargo. The $C^B_{d(v)k}$ parameters were estimated by retrieving the distance between each node in $\hat{\mathcal{N}}_v$ for vessel v and each region k's centroid. The centroid was manually defined to be at an approximate offshore center of all ports associated with a region. From this, the sailing time of vessel v was calculated and multiplied with the unit price of the bonus bunker, $\widetilde{P}$, to estimate the bunker cost of repositioning. Then, any incurred canal costs and ports costs were added to the bunker cost to generate the resulting reposition cost $C^B_{d(v)k}$.

A final group of parameters was randomly sampled from a sensible interval to refrain from creating biasedly constructed test instances. These include the time when vessels first become available to transport cargoes, $\underline{T}_{o(v)v}$, the beginning of the time windows for pickup ports, $\underline{T}_{iv}, i \in \mathcal{N}^P_v$, the initial bunker level on board the vessels, B^0_v, and the number of vessels to be allocated in each region in each scenario, R^K_{ks}. The $\underline{T}_{o(v)v}$ parameters were decided by sampling uniformly from an interval between $[0, 10]$ days. Similarly, the B^0_v parameters were sampled uniformly from the interval $[2\underline{B}_v, \overline{B}_v]$. The lower limit was increased to ensure vessel would not start their schedules with an empty tank. For pickup ports, the upper time windows were constructed by adding ten days to the start. Typically, ten days is what one would experience as an operator in the dry bulk operator industry (Husby, 2022). For the delivery ports, the start of the time windows, $\underline{T}_{iv}, i \in \mathcal{N}^D_v$, were constructed to be the direct sailing time, T^S_{ijv}, plus the start of the time window for the corresponding pickup port, $\underline{T}_{iv}, i \in \mathcal{N}^P_v$. However, the end of the time windows for the delivery ports, $\overline{T}_{iv}, i \in \mathcal{N}^D_v$, were defined to be the time of the end of the time window of the corresponding pickup port, $\overline{T}_{iv}, i \in \mathcal{N}^P_v$, plus twice the direct sailing time, T^S_{ijv}. Increasing the time windows for delivery nodes allows vessels to stop for bunker, or to deviate to take on parceled cargo as is often done in Western Bulk's operational environment.

The R^K_{ks} parameters were generated by starting from a projected allocation of vessels across regions provided by industry experts. For example, Western Bulk might want to allocate five vessels in the first region, three vessels in the second region, ten vessels in the third region, and so on. Denote this projected allocation $\mathcal{R}^* = \{R^*_k \quad \forall k \in \mathcal{K}\}$, where R^*_k denotes the number of vessels allocated in region k in the projected allocation of vessels. For each R^*_k define lower bounds, $\underline{R}^*_k = \lfloor \frac{1}{2} R^*_k \rfloor$, and upper bounds, $\overline{R}^*_k = \lceil \frac{3}{2} R^*_k \rceil$. Then, define a sequence, $\mathcal{S}^*_k = \{\underline{R}^*_k, \underline{R}^*_k + 1, \dots, \overline{R}^*_k - 1, \overline{R}^*_k\}$. For each $\mathcal{S}^*_k \quad \forall k \in \mathcal{K}$, fit a symmetric binomial distribution, $P^*_k = \text{binom}_k(n = |\mathcal{S}^*_k|, p = 0.5)$. For each scenario, $s \in \mathcal{S}$, and region $k \in \mathcal{K}$, sample from P^*_k to generate R^K_{ks}. If $\sum_{k \in \mathcal{K}} R^K_{ks} \neq |\mathcal{V}|$ for a scenario s, discard the sampled parameters and repeat. Here, $|\mathcal{V}|$ is the number of vessels in the fleet. The probability of scenario s, P_s, was set to be equal across scenarios. The set of scenarios, $\mathcal{S}$, was determined by the in-sample stability analysis presented in Section 9.1. Finally, the cargo-related sets such as $\mathcal{N}^P$,$\mathcal{N}^D$,$\mathcal{N}^O$,$\mathcal{N}^C$ and the set of regions $\mathcal{K}$ are specific to each test instance. Test instance generation is further explained in Section 8.2.

Table 8.2 and Table 8.3 summarize the sets and parameters used in the model, respectively, and explain how they were generated.

Sets	Meaning	Provided by
$\mathcal{V}$	Set of vessels	Western Bulk
$\mathcal{B}$	Set of bunker nodes	Industry experts
$\mathcal{N}$	Set of cargo-related nodes	Western Bulk
$\mathcal{K}$	Set of regions	Western Bulk
$\mathcal{S}$	Set of scenarios	Stability Analysis (Section 9.1)

Table 8.2: Sets used in the model and how they were generated

Parameter	Meaning	Depends on	Generated	Provided by
T^S_{ijv}	Sailing time from node i directly to node j for vessel v	Distance	No	Maritime Optima
		Vessel speed	Yes	Industry experts
		Time spent bunkering	Yes	Industry experts
T^Q_{iv}	Time required to load or discharge one unit of cargo at node i with vessel v		Yes	Industry experts
C_{ijv}	Cost of visiting node i and sailing directly from node i to node j, cost of bunker not included	Port costs	Yes	Industry experts
		Canal costs	No	Wilhelmsen (2022)
P^B_i	Price of purchasing one unit of bunker at bunker node i		No	BunkerEx (2022)
$\widetilde{P}$	Unit price of bonus bunker	$\widetilde{P} = \frac{\sum_i^{\lvert\mathcal{B}\rvert} P^B_i}{\lvert\mathcal{B}\rvert}$	No	Calculation
C^S_{ijv}	Cost of servicing the cargo at node i with a spot ship	Spot ship price	No	Fearnleys (2022)
		Spot ship port	Yes	Western Bulk
		Distance	No	Maritime Optima
		Vessel speed	Yes	Industry experts
B^S_{ijv}	Total bunker consumption for vessel v while sailing directly from node i to node j		Yes	Industry experts
B^P_{iv}	Port bunker consumption for vessel v while in node i		Yes	Industry experts
B^0_v	Initial bunker level on board vessel v		Yes	Random sampling
R_i	Revenue generated from transporting one unit of cargo from node i		No	Western Bulk
$\overline{T}_{iv}$	The latest time at which vessel v may begin its service at node i		Yes	Random sampling
$\underline{T}_{iv}$	The earliest time at which vessel v may begin its service at node i		Yes	Random sampling
$\overline{B}_v$	Maximum bunker level for vessel v		Yes	Industry experts
$\underline{B}_v$	Minimum bunker level for vessel v		Yes	Industry experts
$\overline{Q}_i$	Maximum quantity of the cargo at node i to be transported	Cargo quantity transported	Yes	Western Bulk
$\underline{Q}_i$	Minimum quantity of the cargo at node i to be transported	Cargo quantity transported	Yes	Western Bulk
K_v	Cargo carrying capacity of vessel v		No	Western Bulk
$o(v)$	Origin port of vessel v		Yes	Western Bulk
R^K_{ks}	Number of vessels to allocate to region k in scenario s		Yes	Random sampling
$C^B_{d(v)k}$	Reposition cost from vessel v's last visited node to region k	Distance	No	Maritime Optima
		Vessel speed	Yes	Industry experts
		$\widetilde{P}$	No	BunkerEx (2022)
		Canal Cost	No	Wilhelmsen (2022)
		Port Cost	Yes	Industry experts
P_s	Probability of scenario s	$P_S = \frac{1}{\lvert\mathcal{S}\rvert}$	No	Calculation

Table 8.3: Parameters used in the model and how they were generated

8.2 Test Instances

After compiling the sets and parameters, as explained in Section 8.1, a test instance generator was created. By taking on generator parameters such as `NUM_SPOT`, signifying the number of spot cargoes in the problem, `NUM_COA`, denoting the number of CoA cargoes in the problem, and `NUM_VESSELS`, specifying the number of vessels in the problem, `NUM_SCENARIOS`, specifying the number of scenarios, and a `RANDOM_STATE` parameter for reproducibility, a large variety of test instance could be generated. The initial dataset of 9,200 voyages was filtered down to voyages that commenced after January 1st, 2022, as spot ship rates from Fearnleys (2022) were collected for this time period. Western Bulk expressed an interest in the Indian and West-Pacific Oceans (Husby, 2022), so the dataset was further filtered to include voyages that started and ended in these oceans. As such, a set of bunker ports specific to these regions was also constructed. After filtering further for the Supramax and Ultramax subsegment, vessels hired on longer TC contracts, and removing voyages between ports that could not be mapped to Maritime Optima's database, approximately 100 voyages remained. The generator randomly samples the compiled set of 100 voyages to create test instances according to the abovementioned generator parameters. The set of regions $\mathcal{R}$ is generated for each test instance by finding all unique regions of the origin ports for each vessel and the ports associated with each pickup node, delivery node, and bunker node. Figure 8.4 shows an example test instance consisting of five CoA cargoes, ten spot cargoes, five vessels, and five bunker nodes.

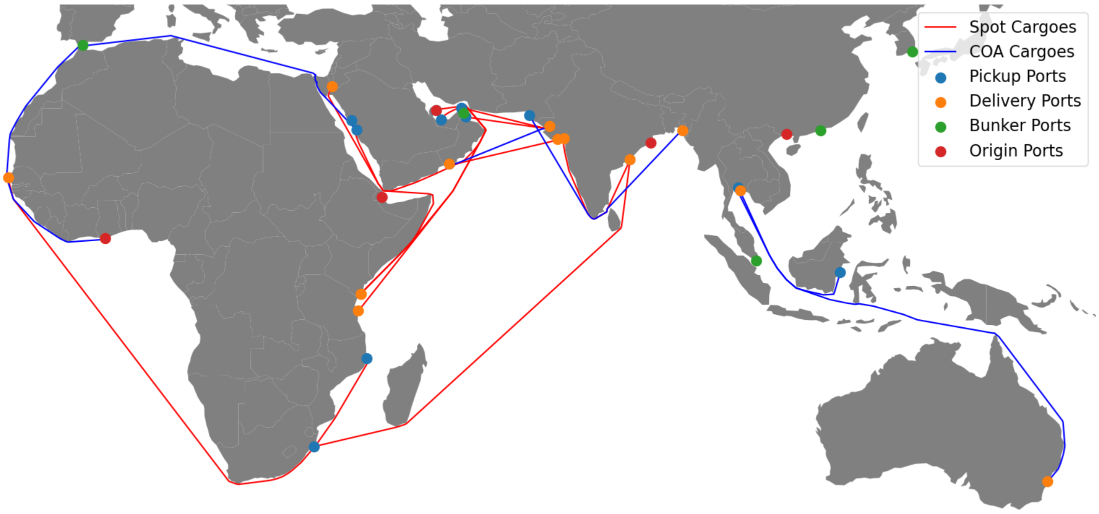


Figure 8.4: An example test instance

For the generated test instances, the number of spot cargoes was fixed to be twice the amount of CoA cargoes. As such, for a test instance of 15 cargoes, there would be five CoA cargoes and ten spot cargoes. According to Husby (2022), this is representable for Western Bulk's operational environment. The MoLOO flexibility limits were fixed to ± 10% for all the generated test instances. The number of cargoes, number of vessels, and number of bunker ports categorize the generated test instances into classes. For example, `C15V3B4` refers to the class of test instances containing 15 cargoes, three vessels, and four bunker nodes. Five different test instances were generated for each class by changing the `RANDOM_STATE` parameter. Thus, test instances differ as they are randomly sampled from about 100 actual voyages performed by Western Bulk. The `RANDOM_STATE` also controls the values of those parameters generated through random sampling. Table 8.4 shows the different classes of generated test instances.

Name	Size	# Cargoes	# Vessels	# Bunker nodes
C9V3B4	Small	9	3	4
C12V3B4	Small	12	3	4
C12V4B6	Small	12	4	6
C15V5B10	Medium	15	5	10
C30V5B10	Medium	30	5	10
C30V10B10	Medium	30	10	10
C60V10B10	Medium	60	10	10
C45V15B10	Medium	45	15	10
C60V15B10	Large	60	15	10
C60V20B10	Large	60	20	10
C90V20B10	Large	90	20	10
C90V30B10	Large	90	30	10
C120V30B10	Large	120	30	10

Table 8.4: Classes of generated test instances

8.3 Implementation

This section highlights implementation considerations for the path flow model presented in Chapters 6, the Adaptive Large Neighborhood Search for the Vessel Combination Problem (ALNS-VCP) outlined in Chapter 7. Section 8.3.1 explains preprocessing techniques leveraged during the construction of networks $(\hat{\mathcal{N}}_v, \hat{\mathcal{A}}_v)$ for each vessel v. Section 8.3.2 highlights caching techniques leveraged during the ALNS-VCP matheuristic to prevent performing repeated operations.

8.3.1 Arc Reductions

As explained in Section 5.2, each vessel v is assigned a network $(\hat{\mathcal{N}}_v, \hat{\mathcal{A}}_v)$. During the node sequence generation presented in Section 6.3.1, these graphs are exhaustively searched by a modified Depth-First-Search (DFS) algorithm. A naïve approach would be to generate all $\hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v$ arcs for each graph and perform the DFS algorithm on these. The number of directed arcs in a fully connected graph grows according to $O(\hat{\mathcal{N}}_v^2)$. The time complexity of the DFS algorithm is proportional to the number of nodes and arcs in the graph. As such, it is necessary to make the $(\hat{\mathcal{N}}_v, \hat{\mathcal{A}}_v)$ graphs as small as possible for the DFS algorithm to complete its search for realistically sized problems in a reasonable amount of time. This section elaborates on some of the arc reduction techniques leveraged to minimize the number of arcs.

The most obvious eliminations exclude arcs that either go to the origin node, $o(v)$, or leave the destination node, $d(v)$. As these nodes are either the starting or ending point of a vessel's schedule, such arcs should be eliminated. Thus, it is possible to eliminate all arcs $(i, o(v)) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v$ and $(d(v), j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v$.

As ports and nodes are not related, traveling to a bunker node in the same port as a pickup node or delivery node is acceptable, but traveling to the same node is not allowed. Thus, arcs $(i, j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | i = j$ should be removed.

Further arc eliminations are achievable by remembering that a vessel has to deliver its cargo before completing its route. As such, a vessel should not be allowed to travel from a pickup node to its destination node. In other words, all arcs $(i, d(v)) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | i \in \mathcal{N}_v^P$ can be eliminated. Following an analogous logic, vessels should not be allowed to travel from their origin node $o(v)$ to a delivery node. Thus, it is possible to eliminate all arcs $(o(d), j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | j \in \mathcal{N}_v^D$. The pickup node precedence logic also facilitates the removal of arcs from delivery nodes to pickup nodes. As such, all arcs $(N + i, i) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | i \in \mathcal{N}_v^P$ should be eliminated.

Next, it would be advisable to remember that a vessel may not travel between two pickup nodes if the sum of the minimum MoLOO limits, $\underline{Q}_i + \underline{Q}_j$, at the pickup nodes is greater than the cargo carrying capacity of vessel v, K_v. Thus, it is possible to remove all arcs $(i, j) \in \hat{\mathcal{N}}_v \times$

$\hat{\mathcal{N}}_v | i \in \mathcal{N}_v^P \wedge j \in \mathcal{N}_v^P \wedge \underline{Q}_i + \underline{Q}_j > K_v$. Using the same logic, vessels may not travel between two delivery nodes if their corresponding pickup nodes have $\underline{Q}_i + \underline{Q}_j > K_v$. As such, all arcs $(i,j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | i \in \mathcal{N}_v^D \wedge j \in \mathcal{N}_v^D \wedge \underline{Q}_{i-N} + \underline{Q}_{j-N} > K_v$ should be removed.

Time windows can also be used to eliminate further arcs. A vessel may not travel to a node with earlier time windows. However, by including the sailing time, T_{ijv}^S and minimum time spent in port $T_{iv}^Q \underline{Q}_i$, more arcs can be eliminated. For pickup nodes, eliminate all arcs $(i,j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | \underline{T}_{iv} + T_{iv}^Q \underline{Q}_i + T_{ijv}^S > \overline{T}_{jv}$. For delivery nodes, eliminate all arcs $(i,j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | \underline{T}_{iv} + T_{iv}^Q \underline{Q}_{i-N} + T_{ijv}^S > \overline{T}_{jv}$. For all remaining nodes, eliminate all arcs $(i,j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | \underline{T}_{iv} + T_{ijv}^S > \overline{T}_{jv}$.

Further, by combining information about time windows with precedence logic, it is possible to remove all arcs $(i,j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | i \in \hat{\mathcal{N}}_v^P \wedge \underline{T}_{iv} + T_{ijv}^S + T_{j,i+N,v}^S > \overline{T}_{i+N,v}$. Figure 8.5 illustrates the situation of a cargo i being picked up in port A to be delivered in port C. Here, going from port A to port B and then to port C will lead to a violation of the time window constraints in port C. As such, the arc (A, B) should be excluded. A similar argument can be made with respect to delivery ports by eliminating all arcs $(i,j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | j \in \hat{\mathcal{N}}_v^D \wedge \underline{T}_{j-N,v} + T_{j-N,iv}^S + T_{ijv}^S > \overline{T}_{jv}$. Figure 8.5 illustrates the situation where arc (B, C) is eliminated due to time violations in port C.

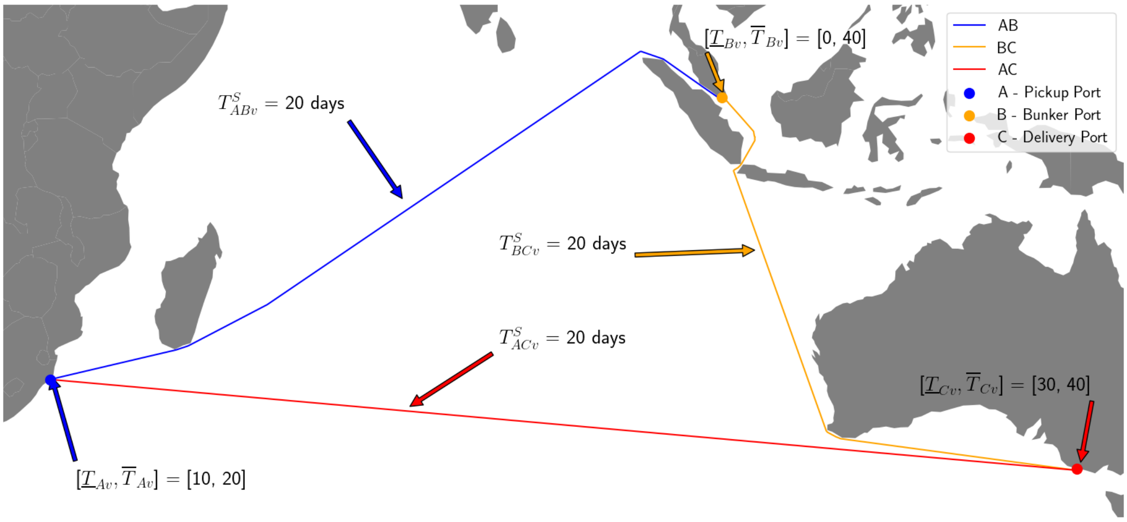

$[\underline{T}_{Bv}, \overline{T}_{Bv}] = [0, 40]$
AB
BC
AC
A - Pickup Port
B - Bunker Port
C - Delivery Port
$T_{ABv}^S = 20$ days
$T_{BCv}^S = 20$ days
$T_{ACv}^S = 20$ days
$[\underline{T}_{Cv}, \overline{T}_{Cv}] = [30, 40]$
$[\underline{T}_{Av}, \overline{T}_{Av}] = [10, 20]$

Figure 8.5: Example where arcs (A, B) and (B, C) can be excluded

Finally, traveling from a bunker node to a different bunker node is not allowed in the TSRSPBO studied in this thesis. Thus, all arcs $(i,j) \in \hat{\mathcal{N}}_v \times \hat{\mathcal{N}}_v | i \in \mathcal{B} \wedge j \in \mathcal{B}$ are removed.

8.3.2 Caching

As shown in Algorithm 5, each potential route must be checked for feasibility with respect to Equations (5.8) - (5.26). If feasible, the profit of each route must be calculated by solving the linear programming model presented in Section 6.3.2. As the ALNS-VCP matheuristic often produces routes previously encountered during the search, storing the result of these relatively computationally heavy operations is beneficial. As such, all infeasible routes for each vessel are stored in a lookup map. Similarly, each vessel's feasible routes and profits are stored in a separate lookup map.

A similar technique is used for the CROSS-Search operator presented in Section 7.4.1. Each combination of two vessels and their respective routes in the current solution are stored in a lookup map. If the combination is encountered later, the CROSS-Search operator is skipped.

Chapter 9

Computational Study

The path flow model and the Adaptive Large Neighborhood Search for the Vessel Combination Problem (ALNS-VCP) were implemented in `C++` leveraging the Gurobi Optimization, LLC (2022) `C++` interface. In addition to having access to Gurobi, `C++` was chosen as the extensive array manipulation required by the ALNS-VCP matheuristic proved to be more efficient than in other programming languages, such as `Python`. Test runs were run on computing nodes on the Microsoft Azure cloud computing platform. Table 9.1 summarizes the system configuration.

Computer	Microsoft Virtual Machine - Standard E20as v5
Processor	AMD EPYC$^{\text{TM}}$ 7763v (Milan) - 3.2 GHz
RAM	160 GB
Operating System	Linux (ubuntu 20.04)
Gurobi Licence Type	Academic
Gurobi version	10.0.1
C++ version	`c++20`

Table 9.1: Description of hardware and software used for the computational study

According to Western Bulk, it is acceptable to wait for a solution of up to one hour (Husby, 2022). Hence, the ALNS-VCP was not allowed to run for more than 3,600 seconds. Similarly, the path flow solution was stopped after one hour, in which case a solution was not considered to have been found. As the ALNS-VCP exhibits some inherent randomness due to the adaptive selection criteria, the ALNS-VCP matheuristic was run five times on every test instance to approximate the heuristic's average performance.

Section 9.1 provides an in-sample stability analysis to determine the number of scenarios to sample. Section 9.2 outlines how the iterative ALNS-VCP matheuristic was configured and tuned to deliver high-quality solutions. Section 9.3 compares solutions found by the finalized ALNS-VCP matheuristic to the path flow solution method with a priori column generation presented in Chapter 6. Section 9.4 inspects solutions generated by the ALNS-VCP and their characteristics. Section 9.5 shows the impact of considering fleet repositioning. Finally, Section 9.6 emphasizes the impact of dedicating resources to bunker procurement divisions by analyzing how the fleet-specific profit varies with bunker price discounts.

9.1 Stability Analysis

An in-sample stability analysis was conducted using the exact path flow model with a priori column generation to determine an appropriate number of scenarios. The test instance class `C30V10B10` is one of the largest test instance classes the path flow solution approach would be able to solve within one hour. As such, class `C30V10B10` was chosen for the stability analysis. The test instance generation procedure outlined in Section 8.2 generated ten different test instances. For each test instance, the scenario-dependent R^K_{ks} parameters denoting how many vessels to allocate in each region in each scenario were sampled. The R^K_{ks} parameters were sampled ten times from the same probability distribution according to the sampling procedure explained in Section 8.1.2. This procedure was repeated by varying the number of scenarios to model. The parameters sampling procedure was repeated for 1, 3, 5, 10, and 15 scenarios. The results from solving the generated test instances by the path flow solution approach are presented in Table 9.2. The **# Scenarios** column denotes the number of scenarios that were sampled. The $\mathbf{Obj_{PF}}$ column presents the optimal path flow objective value averaged across the ten test instances and ten sampled sets of scenario-dependent parameters. The columns $\mathbf{Time_{AP}[s]}$ and $\mathbf{Time_{PF}[s]}$ denote the averaged time in seconds for the a priori column generation and solving the final path flow model, respectively. The **SD** and **CV[%]** columns denote the standard deviation and coefficient of variation across the test instances and ten sampled sets of scenario-dependent parameters, respectively.

# Scenarios	$\mathbf{Obj_{PF}}$	$\mathbf{Time_{AP}[s]}$	$\mathbf{Time_{PF}[s]}$	SD	CV[%]
1	28,110,505	1,428.7	49.4	253,220	0.90
3	28,046,195	1,428.3	49.9	241,188	0.86
5	28,070,379	1,429.7	49.8	220,758	0.77
10	28,043,056	1,432.0	50.6	224,773	0.80
15	28,035,828	1,433.3	51.4	202,705	0.72
Average tot.	28,061,193	1,430.4	50.2	228,529	0.81

Table 9.2: In-sample stability analysis results

Table 9.2 shows a decreasing trend in the $\mathbf{Obj_{PF}}$, **SD**, and **CV[%]** columns with increased scenarios. Furthermore, the **CV[%]** values are relatively low, and the averaged $\mathbf{Obj_{PF}}$ values decrease only marginally with an increased number of scenarios. These observations suggest that for the chosen class of test instances, the second-stage costs only constitute a small share of the total profit generated by the fleet. The **CV[%]** change is most noticeable from three to five and ten to 15 modeled scenarios. The absolute values in the **SD** column show a significant drop when comparing one scenario to five scenarios. Additionally, as using ten scenarios increases the standard deviation compared to five scenarios, in-sample stability was assumed to be reached when using five scenarios. As such, all further experiments conducted in this chapter use five scenarios to sample from.

9.2 ALNS-VCP Setup

This section presents two approaches in which the ALNS-VCP matheuristic was adjusted to provide high-quality solutions to the TSRSPBO studied in this thesis. Section 9.2.1 outlines how local search and the Vessel Combination Problem (VCP) were utilized to increase the performance of the ALNS-VCP. Section 9.2.2 presents the parameter tuning procedure of the resulting ALNS-VCP configuration.

9.2.1 ALNS-VCP Configuration

During the implementation phase of this thesis, it became evident that the operators employed by the Large Neighborhood Search (LNS) of the ALNS procedure had difficulty finding high-quality solutions alone. Initial test runs indicated the importance of incorporating Local Search (LS) operators and the Vessel Combination Problem (VCP), thus introducing the iterative ALNS-VCP matheuristic. To measure the impact of these extensions, three different heuristic configurations were defined. The first configuration was based on the ALNS procedure without incorporating local search of the VCP. The second configuration was the iterative ALNS-VCP matheuristic without local search. The third configuration was the finalized ALNS-VCP matheuristic, which incorporates local search. The parameters of the ALNS procedure were initially set according to Ropke and Pisinger (2006)'s recommendations, which suggest parameter values that generally perform well across a wide range of problems. Subsequently, five test instances were generated for each instance class from the test instance classes listed in Table 8.4. All three of the different configurations were used to solve them. Furthermore, each test instance was solved five times by all of the configurations.

The parameters of the ALNS procedure used directly from Ropke and Pisinger (2006) are summarized in Table 9.3. Additionally, the number of iterations for the configurations, $I^{ALNS-VCP}$, the number of iterations for each segment of the ALNS procedures, I^{S}, and the number of iterations before the VCP is solved, I^{VCP}, were set to 2,500, 50, and 100, respectively.

Parameter	Value	Description
p^{worst}	3	Randomness parameter for worst removal
p^{Shaw}	6	Randomness parameter for Shaw removal
T^{start}	-	Simulated annealing temperature, set such that the probability of accepting a candidate solution is 50% if the candidate solution is less than %5 worse than the constructed solution
c	0.99975	Simulated annealing *cooling rate*
σ_1	33	ALNS score for finding new best solution
σ_2	9	ALNS score for finding new accepted solution
σ_3	13	ALNS score for finding new solution
$\widetilde{r}$	0.1	ALNS reaction parameter
α	40%	Maximum percentage of current solution to destroy

Table 9.3: Initial parameters of the ALNS procedure used directly from Ropke and Pisinger (2006)

Table 9.4 presents the results of comparing the three different configurations for the generated test instances. Each test instance was solved five times by each of the different configurations. The columns labeled **ALNS**, **ALNS-VCP w/o LS**, and **ALNS-VCP** represent the three configurations. The **Instance Class** column displays the classes of the test instances solved along with the size of the class. The values in the **Gap**[%] columns are calculated based on the best objective value found across the different configurations for each test instance and averaged across the five runs. Therefore, it provides a relative measure of the performance of the configurations. The **Time**[s] columns indicate the time in seconds for each configuration to solve the generated test instances averaged across the five runs.

	ALNS		**ALNS-VCP w/o LS**		**ALNS-VCP**	
Instance Class	**Gap**[%]	**Time**[s]	**Gap**[%]	**Time**[s]	**Gap**[%]	**Time**[s]
S C9V3B4	1.15	5.8	0.03	6.0	0.00	6.3
S C12V3B4	2.31	7.4	1.10	8.2	0.87	8.6
S C12V4B6	2.64	17.5	0.34	18.8	0.00	19.7
M C15V5B10	3.50	35.7	0.11	34.9	0.02	37.2
M C30V5B10	10.41	97.9	1.17	75.4	0.17	84.6
M C30V10B10	10.99	189.1	0.68	163.5	0.32	188.3
M C60V10B10	17.90	855.9	0.93	578.4	0.18	641.0
M C45V15B10	14.53	883.2	0.17	681.3	0.03	827.1
L C60V15B10	18.25	1659.8	0.36	1121.1	0.10	1293.7
L C60V20B10	15.23	2181.4	0.17	1839.6	0.03	2162.6
L C90V20B10	18.52	3600.0	0.68	3366.7	0.11	3493.5
L C90V30B10	15.50	3600.0	0.39	3600.0	0.06	3600.0
L C120V30B10	17.48	3600.0	1.13	3600.0	0.23	3600.0
Average tot.	11.42	1287.2	0.56	1161.1	0.16	1227.9

Table 9.4: Comparison of heuristic configurations before tuning parameters of the ALNS procedure

The key finding from Table 9.4 is the significant disparity in the **Average tot.** gap percentage values. The **ALNS** configuration, on average, achieves solutions with a relative gap of 11.42%. In contrast, the iterative ALNS-based matheuristic configurations, **ALNS-VCP w/o LS** and

ALNS-VCP achieve gaps of 0.56% and 0.16%, respectively. The disparity is noteworthy because the inclusion of the VCP does not directly contribute to finding a wider range of vessel-specific routes. Instead, the VCP enables the search to find combinations of vessel-specific routes encountered by the LNS operators. This observation suggests that the implemented LNS operators used by the ALNS procedure are effective at discovering various vessel-specific routes but struggle to generate well-performing combinations of these routes. Finally, as the **ALNS-VCP** configuration decreases the **Averaged tot.** gap from 0.56% to 0.16%, local search is shown to contribute to the exploration of the search space.

9.2.2 Tuning the parameters of the ALNS procedure

Although Ropke and Pisinger (2006)'s reported the parameter values of the ALNS procedure to work well on a variety of different problems, further parameter tuning specific to the TSRSPBO studied in this thesis is required. To this end, this section presents a parameter tuning analysis for the parameters of the ALNS procedure of the iterative ALNS-VCP matheuristic.

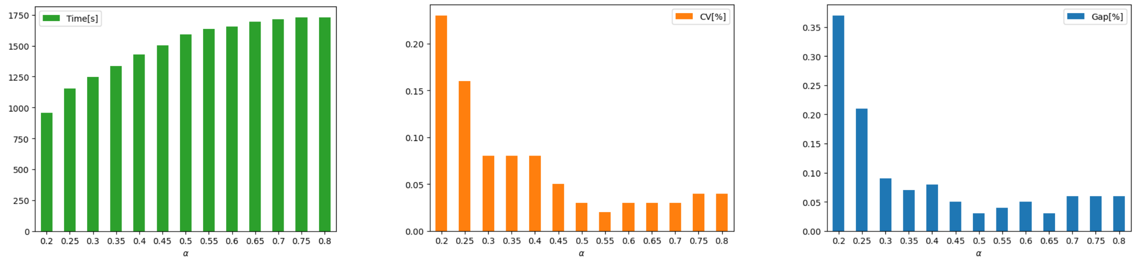
While experimenting with the implementation of the ALNS-VCP matheuristic, we discovered that certain parameter values of the ALNS procedure mentioned in Ropke and Pisinger (2006) consistently generated high-quality solutions and showed minimal sensitivity to changes in their values. However, we observed that the solution quality heavily relied on the percentage of the current solution that is destroyed in the ALNS procedure.

The number of cargoes removed in each iteration of the ALNS-VCP matheuristic is decided by drawing randomly from the discrete uniform distribution $Unif[1, \alpha|\mathcal{T}|]$, where $\alpha \in \mathbb{R} \,|\, 1 < \alpha|\mathcal{T}| \wedge \alpha < 1$ is a constant. Here, $\mathcal{T}$ is the set of cargoes serviced in the current solution. During the early stages of the development of the ALNS-VCP matheuristic, the α parameter was set to have a value of 0.4. As increasing the value of the α parameter seemed to produce higher-quality solutions, a more systematic parameter tuning experiment was deemed necessary.

To avoid the potential issue of overfitting the tuned parameters to a specific class of instances, we generated two test instances for each of the following classes: `C30V5B10`, `C30V10B10`, `C45V15B10`, `C60V20B10`, and `C90V30B10`. These test instances were then solved using the ALNS-VCP, utilizing the parameter values of the ALNS procedure shown in Table 9.3, with the exception of the α value. $I^{ALNS-VCP}$, I^S and I^{VCP} were set to 2,500, 50, and 100, respectively. Each test instance was solved five times due to the inherent randomness in the ALNS-VCP matheuristic. Initially, the α value was set to 0.2 and subsequently increased by increments of 0.05 until reaching a final value of 0.8. The results of tuning the α parameter value are shown in Table 9.5. The α rows contain the different α values used in the tuning procedure. The **Time[s]** rows show the averaged time in seconds used to solve the generated test instances. The **Gap[%]** rows display the relative gap measured across the different values of α averaged across the number of runs and generated test instances. Finally, the **CV[%]** rows present the coefficient of variation measured across the five runs for each test instance and averaged across each instance. The results are further visualized in Figure 9.1.

α	**0.2**	**0.25**	**0.3**	**0.35**	**0.4**	**0.45**	**0.5**	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**
Time[s]	958	1,153	1,248	1,337	1,430	1,504	1,592	1,634	1,655	1,696	1,716	1,728	1,731
Gap[%]	0.37	0.21	0.09	0.07	0.08	0.05	0.03	0.04	0.05	0.03	0.06	0.06	0.06
CV[%]	0.23	0.16	0.08	0.08	0.08	0.05	0.03	0.02	0.03	0.03	0.03	0.04	0.04

Table 9.5: Results of tuning the α value for the ALNS-VCP matheuristic



(a) Tuning time vs. α

(b) Tuning CV vs. α

(c) Tuning gap vs. α

Figure 9.1: Tuning results vs. α

As anticipated, Figure 9.1a illustrates that the time required to solve the generated test instances increases as the value of α increases. This finding makes sense since the repair operators in the ALNS procedure need to insert, on average, more cargoes. Figure 9.1b demonstrates a decreasing trend in the coefficient of variation across the ALNS-VCP matheuristic tuning runs when averaged across the test instances. This trend continues until an α value of 0.55, beyond which the coefficient of variation starts to rise. Figure 9.1c shows a similar pattern, reaching a minimum gap value of 0.03 at an α value of 0.5. However, it is worth noting that an α value of 0.65 also achieves an equivalent gap.

Based on Table 9.5 and Figure 9.1, an α value of 0.5 was determined to achieve the best results. Although an α value of 0.65 achieves an equivalent gap, such a value increases the solution time and the coefficient of variation.

9.3 Comparison of Solution Methods

This section presents a comparison between the solutions produced by the iterative ALNS-VCP matheuristic and the path flow solution method with a priori column generation. Five test instances for each of the classes presented in Table 8.4 were generated. Each test instance was solved a total of five times by the ALNS-VCP heuristic. The path flow solution method was limited to generating columns for up to one hour, after which a solution was not considered to have been found. It should be noted that increasing the path flow solution time for more than one hour would lead to situations in which the computer would run out of memory. The ALNS-VCP matheuristic was similarly run for up to 3,600 seconds, after which the best-encountered solution was returned.

Table 9.6 summarizes the results. The **Instance Class** column shows the solved classes of test instances along with their size. The $\mathbf{Obj_{PF}}$ presents the objective value found by solving the test instances by the path flow solution method with a priori column generated. The values are averaged across the test instances in each class. The $\mathbf{Time_{AP}}$**[s]** column displays the averaged duration of the a priori column generation. Similarly, the $\mathbf{Time_{PF}}$**[s]** gives the averaged duration of solving the resulting path flow model. $\mathbf{Obj_{AV}}$ gives the objective value found by solving the test instances by the ALNS-VCP matheuristic. The values are averaged across the five runs and test instances in each instance class. Similarly, the $\mathbf{Time_{AV}}$**[s]** column shows the averaged solution time of the ALNS-VCP matheuristic. The **CV[%]** column presents the coefficient of variation calculated across the five ALNS-VCP runs of each test instance. The values are then averaged across the test instances in each instance class. Finally, the **Gap[%]** column displays the averaged absolute percentage gap between the exact solution found by the path flow solution method with a priori column generation and the ALNS-VCP matheuristic. The values are rounded to two decimals.

Instance Class	**Obj$_{\mathbf{PF}}$**	**Time$_{\mathbf{AP}}$**[s]	**Time$_{\mathbf{PF}}$**[s]	**Obj$_{\mathbf{AV}}$**	**Time$_{\mathbf{AV}}$**[s]	**CV**[%]	**Gap**[%]
S C9V3B4	6,553,546	0.6	0.03	6,553,371	6.6	0.00	0.00
S C12V3B4	8,285,835	2.2	0.04	8,285,702	9.3	0.00	0.00
S C12V4B6	10,404,761	8.4	0.11	10,404,761	18.2	0.00	0.00
M C15V5B10	12,056,669	81.0	1.12	12,055,775	39.9	0.01	0.01
M C30V5B10	13,762,712	799.2	16.37	13,759,422	128.0	0.00	0.02
M C30V10B10	27,517,394	1,916.8	40.32	27,515,921	335.5	0.00	0.01
M C60V10B10	-	3,600.0	-	31,990,297	783.2	0.13	-
M C45V15B10	-	3,600.0	-	46,402,247	1,072.8	0.03	-
L C60V15B10	-	3,600.0	-	50,196,997	1,773.1	0.03	-
L C60V20B10	-	3,600.0	-	62,489,404	3,151.4	0.01	-
L C90V20B10	-	3,600.0	-	71,280,724	3,540.4	0.13	-
L C90V30B10	-	3,600.0	-	98,877,545	3,600.0	0.05	-
L C120V30B10	-	3,600.0	-	102,861,735	3,600.0	0.22	-
Average tot.	-	2,154.5	-	41,744,146	1,389.1	0.05	-

Table 9.6: Comparison of solution approaches

The results presented in Table 9.6 demonstrate the effectiveness of the different solution approaches. The path flow method, combined with a priori column generation, successfully solves all small test instances optimally. However, it only achieves optimal solutions for three out of five medium-sized instances and none of the large instances. In contrast, the ALNS-VCP matheuristic is capable of solving all small and medium-sized instances within an hour. Among the large instances, the ALNS-VCP matheuristic solves all test instances of the C60V15B10 and C90V20B10 classes within an hour, while instances in the other large classes reach the one-hour stopping criterion.

The ALNS-VCP matheuristic achieves optimal solutions for all small-sized C12V4B6 instances. For the remaining small instances, the optimality gaps are less than 0.005, which are considered negligible and rounded to zero. In the first three medium-sized classes, the ALNS-VCP reports average optimality gaps of 0.01, 0.02, and 0.01, respectively, indicating high-quality solutions. Since the path flow approach fails to solve larger instances, no optimality gaps are reported for them. However, the ALNS-VCP provides solutions for all generated test instances. The **CV**[%] column values remain relatively low for larger instances, indicating that the ALNS-VCP matheuristic produces solutions of similar high-quality across multiple runs, even for larger-sized test instances.

Furthermore, a comparison of solution times reveals interesting insights. For small-sized instances, the path flow approach is advantageous, providing optimal solutions in a shorter solution time. However, for the C15V5B10, C30V5B10, and C30V10B10 classes, the ALNS-VCP delivers near-optimal solutions with significantly reduced computation times of 52%, 84%, and 83%, respectively.

Overall, when comparing the ALNS-VCP matheuristic to the path flow approach, it is evident that the path flow approach cannot solve optimally or even produce feasible solutions to larger instances. In contrast, the ALNS-VCP consistently produces high-quality solutions within the one-hour time limit, making it a practical operational support tool for dry bulk operators like Western Bulk, even for realistic-sized problems.

9.4 Inspection of Solutions

This section provides some details of the solutions found when solving the generated test instances in the chosen instance classes. Table 9.7 provides cargo-related statistics that are averaged across the test instances and the number of runs for each instance class. The **Instance Class** column shows the instance classes and their size. The $|\boldsymbol{\mathcal{N}}^{\boldsymbol{C}} \in \boldsymbol{\mathcal{T}}|$ column shows the averaged number of contracted Contract of Affreightment (CoA) cargoes serviced by vessels in the fleet in the found solutions. The $|\boldsymbol{\mathcal{N}}^{\boldsymbol{O}} \in \boldsymbol{\mathcal{T}}|$ column gives the averaged number of optional spot cargoes serviced by the fleet. The $|\boldsymbol{\mathcal{N}}^{\boldsymbol{C}} \in \boldsymbol{\mathcal{U}}|$ column displays the averaged number of contracted CoA cargoes serviced by hired spot vessels. Finally, the $|\boldsymbol{\mathcal{N}}^{\boldsymbol{O}} \in \boldsymbol{\mathcal{U}}|$ shows the averaged number of optional spot cargoes that are not serviced by any vessel in the fleet. The averaged values are rounded to the closest integer value.

Instance Class	$\lvert \mathcal{N}^C \in \mathcal{T} \rvert$	$\lvert \mathcal{N}^O \in \mathcal{T} \rvert$	$\lvert \mathcal{N}^C \in \mathcal{U} \rvert$	$\lvert \mathcal{N}^O \in \mathcal{U} \rvert$
S C9V3B4	2	3	1	3
S C12V3B4	3	3	1	5
S C12V4B6	4	5	0	3
M C15V5B10	4	6	1	4
M C30V5B10	7	6	3	14
M C30V10B10	9	14	1	6
M C60V10B10	13	13	7	27
M C45V15B10	14	22	1	8
L C60V15B10	18	21	2	19
L C60V20B10	19	32	1	8
L C90V20B10	26	29	4	31
L C90V30B10	29	46	1	14
L C120V30B10	35	43	5	37
Average tot.	14	19	2	14

Table 9.7: Averaged statistics of solutions

Table 9.7 highlights the importance of servicing the contracted CoA cargoes by vessels in the fleet. In the best-found solutions, the number of spot vessels used to service CoA cargoes is significantly lower than the number of hired spot vessels. However, apart from the C12V3B4 instance class, the number of hired spot vessels is greater than zero. This observation emphasizes the benefit of the hiring optionality of spot vessels. There may very well exist situations in which highly profitable spot cargoes are available. In these situations, the spot cargoes should be prioritized over less profitable CoA cargoes, even though the hire of a spot vessel is required.

Table 9.8 provides relevant bunker-related statistics for the best-found solutions. The **Instance Class** column is defined as before. The **Tot. # Bunker** column shows the total number of bunker nodes visited in the solutions. The values are averaged across the test instances and the number of runs in each instance class and rounded to the closest integer. Similarly, the **Avg. # Bunker (v)** displays the number of bunker nodes visited on average by each vessel in the fleet in the best-found solutions. The values are rounded to one decimal. The **Max. # Bunker (v)** provides the number of bunker node visits for the vessel with the most visits across the test instances and runs in each instance class.

Instance Class	**Tot. # Bunker**	**Avg. # Bunker (v)**	**Max # Bunker (v)**
S C9V3B4	3	1.0	2
S C12V3B4	3	1.0	2
S C12V4B6	4	1.0	2
M C15V5B10	8	1.6	3
M C30V5B10	7	1.4	3
M C30V10B10	15	1.5	3
M C60V10B10	15	1.5	3
M C45V15B10	23	1.5	3
L C60V15B10	22	1.5	3
L C60V20B10	28	1.4	3
L C90V20B10	30	1.5	3
L C90V30B10	41	1.4	3
L C120V30B10	41	1.4	3
Average tot.	18	1.3	2.8

Table 9.8: Averaged bunker node statistics of solutions

For the small instance classes, Table 9.8 shows that the best-found solutions contain routes where each vessel visits a single bunker node on average. For the same instance classes, the maximum number of bunker visits is two. For the medium and large instance classes, the average number of bunker node visits is significantly higher and remains around 1.5 across the instance classes of these sizes. The **Max # Bunker (v)** column displays a similar trend. In general, the number of bunker node visits depends on the planning horizon of the problem. In the problem studied in this thesis, the planning horizon is usually less than 80 days. Table 9.8 shows that when utilizing such planning horizons, one can expect that the maximum number of bunker nodes visited by a vessel should not exceed three.

9.5 Impact of Fleet Repositioning

The Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) studied in this thesis incorporates the extension of fleet repositioning. To quantify the impact of fleet repositioning, four instance classes were chosen to study further. Five test instances from the `C30V10B10`, `C45V15B10`, `C60V20B10`, and `C90V30B10` instance classes were generated. Each test instance was solved five times by the ALNS-VCP matheuristic.

In the first experiment, the test instances were first solved by only considering the first-stage phase of the two-stage stochastic TSRSPBO presented in Chapter 5. As such, fleet repositioning is not considered. The first-stage solutions were then compared to the two-stage stochastic solutions by calculating and subtracting the resulting recourse cost. As described in Section 8.1.2, the repositioning costs on which the recourse cost depends are calculated based on bunker costs and canal costs. Table 9.9 summarizes the results of comparing first-stage solutions to the two-stage stochastic solutions with five scenarios.

The **Instance Class** column shows the instance classes and their size. The $\mathbf{Obj_{Det}}$ column shows the objective values obtained by solving the deterministic first-stage phase of the problem without considering fleet repositioning. The values are averaged over the five runs and five test instances in each instance class. The $\mathbf{Obj_{RC}}$ shows the averaged objective values of $\mathbf{Obj_{Det}}$ after subtracting the second-stage recourse costs. The $\mathbf{Time_{Det}}$**[s]** column highlights the averaged solution time of the deterministic first-stage problems. The $\mathbf{Obj_{SS}}$ presents the averaged objective values of the two-stage stochastic solutions. The $\mathbf{Time_{SS}}$**[s]** gives the averaged solution time for the two-stage stochastic problems. Finally, the **Gap[%]** column shows the averaged gap between the first-stage solutions less the resulting second-stage recourse cost ($\mathbf{Obj_{RC}}$) and the stochastic solutions ($\mathbf{Obj_{SS}}$).

Instance Class	$\mathbf{Obj_{Det}}$	$\mathbf{Obj_{RC}}$	$\mathbf{Time_{Det}}$**[s]**	$\mathbf{Obj_{SS}}$	$\mathbf{Time_{SS}}$**[s]**	**Gap[%]**
`M C30V10B10`	29,710,147	28,495,880	250.6	28,824,484	290.2	1.14
`M C45V15B10`	46,457,344	44,222,937	831.3	44,854,549	971.1	1.41
`L C60V20B10`	65,733,554	63,842,667	2469.9	64,384,387	2721.1	0.84
`L C90V30B10`	103,301,235	99,570,307	3600.0	100,524,767	3600.0	0.95
Average tot.	61,300,571	59,032,948	1,788.0	59,647,047	1,895.6	1.08

Table 9.9: Objective value differences between first-stage and two-stage stochastic solutions when repositioning cost is based on bunker costs and canal costs.

Table 9.9 indicates that the first-stage objective values presented in $\mathbf{Obj_{Det}}$ column are significantly higher than the objective values obtained from solving the two-stage stochastic model presented in $\mathbf{Obj_{SS}}$. However, as the first-stage solutions are found without considering the repositioning cost, these costs can be quite substantial, as indicated by column $\mathbf{Obj_{RC}}$. When comparing the two approaches, solving the two-stage stochastic problem formulation yields, on average, a 1.08% increase in the objective function value for the 20 test instances in the `C30V10B10`, `C45V15B10`, `C60V20B10`, `C90V30B10` instance classes

In the second experiment, rather than ignoring the cost of repositioning, a single scenario was generated based on the expected value of the scenarios used in the full two-stage stochastic problem formulation. The resulting problem is denoted as the deterministic mean value problem. The solutions obtained from solving the deterministic mean value problem were compared to the two-stage formulation with five scenarios. To do so, the recourse cost of the deterministic mean value problem solution was calculated based on solving the second-stage of the stochastic problem formulation with five scenarios. The resulting recourse cost was then subtracted from the first-stage objective value of the deterministic mean value problem. Table 9.10 highlights the results.

Columns **Instance Class**, $\mathbf{Obj_{SS}}$, and $\mathbf{Time_{SS}}$**[s]** are defined as before. The $\mathbf{Obj_{MVP}}$ column shows the objective values obtained by solving the deterministic mean value problem. The values are averaged over the five runs and five test instances in each instance class. The $\mathbf{Obj_{EEV}}$ columns present the averaged objective values obtained when subtracting the second-stage recourse cost based on five scenarios from the first-stage profit of the deterministic mean value problem. The $\mathbf{Time_{MVP}}$**[s]** columns show the averaged solution times for the two-stage formulation based on the expected scenario. Finally, the **VSS[%]** column shows the averaged values of the stochastic solution as a percentage.

Instance Class	$\mathbf{Obj_{MVP}}$	$\mathbf{Obj_{EEV}}$	$\mathbf{Time_{MVP}}$**[s]**	$\mathbf{Obj_{SS}}$	$\mathbf{Time_{SS}}$**[s]**	**VSS[%]**
`M C30V10B10`	28,936,920	28,348,469	256.6	28,824,484	290.2	1.65
`M C45V15B10`	44,928,976	43,979,907	868.3	44,854,549	971.1	1.95
`L C60V20B10`	64,513,642	63,774,819	2547.9	64,384,387	2721.1	0.95
`L C90V30B10`	100,712,461	98,837,711	3600.0	100,524,767	3600.0	1.68
Average tot.	59,773,000	58,735,227	1,818.2	59,647,047	1,895.6	1.56

Table 9.10: Value of the stochastic solution when repositioning cost is based on bunker costs and canal costs.

Looking at Table 9.10, it is evident that solutions of the deterministic mean value problem account for the expected cost of repositioning as the values of the $\mathbf{Obj_{MVP}}$ column are lower than the values of the $\mathbf{Obj_{Det}}$ column found in Table 9.9. Further, the averaged difference between the values of the $\mathbf{Obj_{MVP}}$ and $\mathbf{Obj_{EEV}}$ columns in Table 9.10 is smaller than the averaged difference between the values of the $\mathbf{Obj_{Det}}$ and $\mathbf{Obj_{RC}}$ columns in Table 9.9. Interestingly, the averaged values of the $\mathbf{Obj_{RC}}$ column in Table 9.9 are greater than the averaged values of the $\mathbf{Obj_{EEV}}$ column in Table 9.10. Consequently, the averaged percentage values in the **VSS[%]** column in Table 9.10 are greater than the values of the **Gap[%]** column in Table 9.9. These observations complement the results presented in Table 9.2, highlighting the importance of modeling more than a single scenario.

As explained in Section 8.1.2, the cost of repositioning a vessel from its last planned port visit to a given region is based on the variable sailing costs, such as the cost of bunker consumed and any potential canal costs. However, during repositioning legs, vessels do not carry cargo. As such, there is an associated opportunity cost of repositioning that may be estimated by a charter rate aggregated over the time it takes to reposition. By including this opportunity cost in the calculation of the $C^B_{d(v)k}$ parameters, a more accurate estimation of the second-stage recourse cost can be obtained.

Tables 9.11 and 9.12 summarize the results obtained from repeating the first two experiments when including the opportunity cost in the estimation of the second-stage recourse cost. The columns are defined similarly as in Table 9.9 and 9.10, respectively.

Instance Class	$\mathbf{Obj_{Det}}$	$\mathbf{Obj_{RC}}$	$\mathbf{Time_{Det}}$**[s]**	$\mathbf{Obj_{SS}}$	$\mathbf{Time_{SS}}$**[s]**	**Gap[%]**
`M C30V10B10`	29,710,147	27,021,840	250.6	28,302,306	255.7	4.52
`M C45V15B10`	46,457,344	41,457,699	831.3	44,151,917	780.2	6.10
`L C60V20B10`	65,733,554	61,536,298	2469.9	63,749,872	2463.4	3.47
`L C90V30B10`	103,301,235	94,895,385	3600.0	99,607,008	3600.0	4.73
Average tot.	61,300,571	56,227,806	1,788.0	58,952,776	1,774.8	4.71

Table 9.11: Objective value differences between first-stage and two-stage stochastic solutions when repositioning cost includes opportunity cost.

Instance Class	$\mathbf{Obj_{MVP}}$	$\mathbf{Obj_{EEV}}$	$\mathbf{Time_{MVP}}$[s]	$\mathbf{Obj_{SS}}$	$\mathbf{Time_{SS}}$[s]	VSS[%]
M C30V10B10	28,873,412	28,093,464	215.3	28,302,306	255.7	0.74
M C45V15B10	44,693,870	44,000,647	656.1	44,151,917	780.2	0.34
L C60V20B10	64,227,590	63,541,429	2270.4	63,749,872	2463.4	0.33
L C90V30B10	100,396,194	99,453,143	3600.0	99,607,008	3600.0	0.15
Average tot.	59,547,767	58,772,171	1,685.5	58,952,776	1,774.8	0.39

Table 9.12: Value of the stochastic solution when repositioning cost includes opportunity cost.

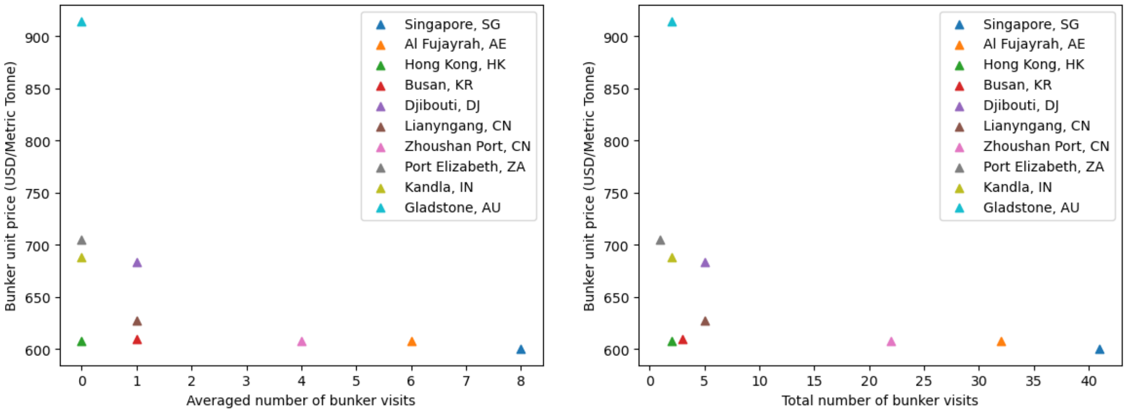
When comparing Table 9.9 with Table 9.11, it is clear that including the opportunity cost when calculating the repositioning cost has a significant impact on the second-stage recourse cost. While the values of the $\mathbf{Obj_{Det}}$ columns are the same in both tables, the $\mathbf{Obj_{RC}}$ column values in Table 9.11 are significantly lower than in Table 9.9, indicating that including the opportunity cost renders the deterministic first-stage solutions less effective. Further, the $\mathbf{Obj_{SS}}$ column values in Table 9.11 are only marginally smaller than those in Table 9.9. As a result, the **Gap[%]** column values are significantly higher in Table 9.11. The **Gap[%]** columns in Tables 9.9 and 9.11 highlight the importance of considering fleet repositioning when finding solutions to TSRSPBOs in the dry bulk industry. If a dry bulk operator does not consider the regions of which vessels are located at the end of the planning horizon, the **Gap[%]** columns in Tables 9.9 and 9.11 indicate that there is a substantial amount of profit that is lost.

In Table 9.12, the $\mathbf{Obj_{MVP}}$ column indicates that solutions of the deterministic mean value problem account for the expected cost of repositioning as the values are lower than the $\mathbf{Obj_{Det}}$ column values in Table 9.11. Further, the $\mathbf{Obj_{EEV}}$ values in Table 9.12 are lower than the $\mathbf{Obj_{EEV}}$ values in Table 9.10, indicating the impact of including the opportunity cost when calculating repositioning costs. Interestingly, including opportunity costs decrease the **VSS[%]** column values when compared to Table 9.10. As such, the deterministic mean value problem performs better when including opportunity costs compared to when opportunity costs are excluded. However, the **VSS[%]** column values in Table 9.12 still indicate the added value of modeling more than a single scenario.

9.6 Impact of Bunker Procurement

As explained in Section 8.1.1, the bunker prices used as input to the model presented in Chapters 5 and 6 were retrieved from Maritime Optima's BunkerEx (2022) interface. These prices are considered market prices and are accessible to anyone seeking to buy bunker fuel at a specific port. However, shipping companies allocate resources to secure favorable bunker purchase agreements in advance. For instance, Western Bulk has a dedicated department responsible for negotiating bunker purchase deals with various ports. This section indicates how the ALNS-VCP can provide decision support for the most effective allocation of resources for such a bunker department.

Figure 9.2 shows the average and total number of visits to bunker ports in the best solutions found for the test instances in the C45V15B10 instance class. The y-axis shows the bunker price, while the x-axis indicates the number of visits to the bunker port. Figure 9.2 shows that Port Elizabeth, Gladstone, Kandla, and Hong Kong ports are rarely visited by vessels. Singapore, Al Fujayrah, and Zhoushan are the most frequently visited bunker ports. This observation can be attributed to the lower bunker prices but also to the location of the ports. The port of Singapore, for example, is positioned strategically between the Indian Ocean and the East China Sea. On the other hand, even though the bunker price in Hong Kong is relatively low, vessels seem to ignore this option. Presumably, traveling to Hong Kong constitutes a detour compared to purchasing bunker in either Singapore or Zhoushan Port.



(a) Averaged number of bunker visits

(b) Total number of bunker visits

Figure 9.2: Number of bunker visits in the C45V15B10 instance class

Figure 9.3 illustrates the average increase in profit for the test instances in C45V15B10 at varying bunker discount rates. The bunker discount rates were applied one at a time to each of the bunker ports. As anticipated, raising the discount rate results in a higher profit. Gladstone and Port Elizabeth, situated in Australia and South Africa, respectively, are positioned inconveniently for vessels operating in the Indo-Pacific region. Furthermore, neither country is a major oil producer, resulting in comparatively higher bunker prices at these ports, as evident in Figure 9.2. Therefore, acquiring a discounted bunker price at these ports is unlikely to significantly contribute to the fleet's overall profit. This notion is confirmed by Figure 9.3, where even with a 20% bunker price discount, the fleet-specific profit only experiences a marginal increase compared to similar discounts at other bunker ports. On the other hand, bunker price discounts at the ports of Singapore or Al Fujayrah have the most significant impact on the average generated profit. This observation aligns with the data presented in Figure 9.2, as these two ports are the most frequently visited and offer the lowest bunker purchase prices. Consequently, if Western Bulk needs to prioritize efforts in securing a discounted bunker purchase contract, Figure 9.3 suggests focusing on either Singapore or Al Fujayrah.

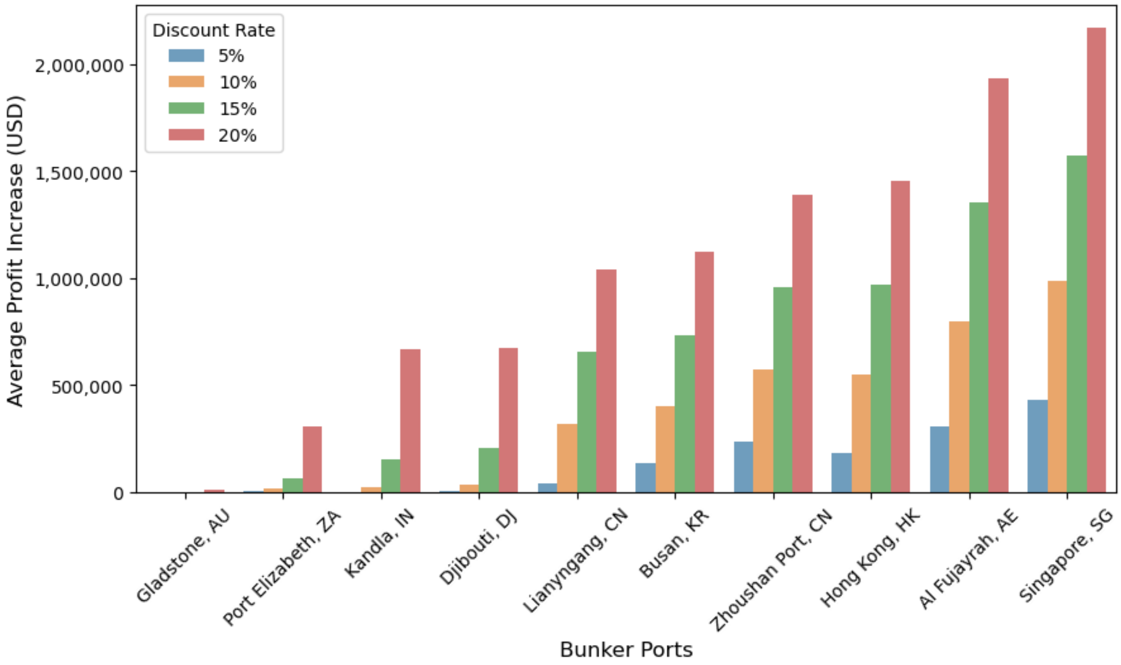


Figure 9.3: Averaged effects of procuring discounted bunker for test instances in C45V15B10

It is interesting that Hong Kong, a port that had infrequent visits in Figure 9.2, has a more significant influence on the average profit for the test instances in `C45V15B10` compared to Zhoushan Port, which was visited relatively more frequently. Although visiting Hong Kong may require a detour when compared to Singapore or Zhoushan, Figure 9.3 indicates that the cost of this detour is minimal compared to the increased profit gained from securing a favorable bunker purchase contract in Hong Kong.

During bunker purchase negotiations, the bunker purchase managers at Western Bulk can use the information provided in Figure 9.3. By analyzing the data, they can identify the significant impact that Al Fujayrah and Singapore have on the profitability of their fleet. As a result, they can prioritize obtaining discounted bunker purchase contracts in these ports. On the other hand, allocating resources to secure favorable contracts at Gladstone and Port Elizabeth would be unnecessary since these ports have a negligible impact on the generated profit. Additionally, if Western Bulk is currently invests resources in securing purchase contracts at Zhoushan Port, it should reconsider its strategy and shift its focus to Hong Kong instead. Such a shift is advisable because a discounted bunker price in Hong Kong would have a greater impact on the company's profits.

Chapter 10

Concluding Remarks

This thesis studies and solves a Tramp Ship Routing and Scheduling Problem with Bunker Optimization (TSRSPBO) defined in collaboration with a dry bulk operator, Western Bulk. The TSRSPBO is modeled as a two-stage stochastic optimization problem and includes three extensions relevant to dry bulk operators. The TSRSPBO concerns generating optimal cargo schedules for a fleet of vessels and a set of contracted and mandatory cargoes to be transported. The objective of the TSRSPBO is to maximize the overall profit generated by the vessels in the fleet. A vessel may transport any amount of cargo within predetermined MoLOO flexibility limits. As such, the optimal amount of cargo to transport must be decided. The problem further incorporates bunker optimization as vessels in the fleet must choose where to bunker and the amount of bunker to purchase. Finally, the TSRSPBO considers regional vessel distributions at the end of the planning horizon.

Western Bulk face TSRSPBOs daily and relies on charting managers to make optimal decisions. This study aims to provide chartering managers with an additional tool to aid their decision-making process. To model Western Bulk's operational environment, an arc flow formulation was proposed based on previous research conducted by Brønmo et al. (2007b), Vilhelmsen et al. (2014), and Omholt-Jensen (2022). The arc flow was further reformulated as a two-stage stochastic optimization model incorporating fleet repositioning costs. The proposed model is the first to combine flexible cargo quantity limits, as studied in Brønmo et al. (2007b), and integrated bunker optimization, as presented in Vilhelmsen et al. (2014), with fleet repositioning considerations.

The generated test instances were based on real-life information from Western Bulk's operational environment. Further, industry partner Maritime Optima provided accurate routing information.

As the arc flow model could not be solved for realistically sized test instances, an exact path flow formulation leveraging a priori column generation was proposed. A modified Depth-First-Search algorithm generates all feasible routes for each vessel in the fleet. A linear programming problem solves each feasible route to optimize the transported cargo amount and purchased bunker amount. Solutions to the linear programming problem yield the optimal vessel-specific profit used in the path flow formulation as columns. The proposed path flow formulation was solved to optimality for test instances of up to 30 cargoes, ten vessels, and ten bunker ports within one hour.

To solve larger test instances, this thesis proposes an iterative matheuristic incorporating an Adaptive Large Neighborhood Search (ALNS) framework for column generation. The resulting columns are fed into a path flow formulation named the Vessel Combination Problem (VCP), which is solved at regular intervals. The complete matheuristic framework is called the Adaptive Large Neighborhood Search for the Vessel Combination Problem (ALNS-VCP).

The computational study suggests that the ALNS-VCP matheuristic finds optimal solutions for small-sized test instances and near-optimal solutions for medium-sized test instances. For the larger test instances, the path flow solution method fails to provide any solution within a reasonable time. In contrast, the ALNS-VCP consistently solves problems of up to 30 vessels, 120 cargoes, and ten bunker nodes within one hour.

Managerial insights provided by the computational study quantify the impact of modeling the TSRSPBO with fleet repositioning as a two-stage stochastic optimization problem. The results confirm that a better regional allocation of vessels could increase fleet-wide profits. The impact of bunker procurement is further investigated using discounted bunker purchase prices. Their effect on the profit generated by the fleet is significant and further provides valuable insights for Western Bulk's bunker procurement division.

Chapter 11

Future Research

This chapter proposes additional areas of research that can be pursued based on the work of the presented thesis. In Section 11.1, potential modifications to the model are suggested in order to better depict the operational environment of a dry bulk operator. Section 11.2 discusses solution methods that could enhance the ability to solve larger test instances. Finally, Section 11.3 outlines additional experiments that can be conducted to quantify the impact of fleet repositioning.

11.1 Model Extensions

Cargo Coupling
As discussed in Section 4.1, the models presented in this thesis do not account for the concept of dirty cargoes, despite it being a characteristic of Western Bulk's operational environment. Therefore, a logical expansion would be to introduce constraints that indicate whether cargoes are coupled and to incorporate a hull cleaning cost for coupled cargoes. Previous research in the field of Ship Routing and Scheduling (SRS) literature, specifically the study by Fagerholt et al. (2013), has explored cargo coupling. Thus, it would be natural to extend the arc flow model by including coupled cargo constraints.

Non-Stationary Bunker Prices
In this thesis, it was assumed that bunker prices remain constant over time. However, considering the dynamic nature of bunker prices would be a natural extension and a more realistic assumption. To incorporate non-stationary bunker prices, the input data can be manipulated by dividing bunker nodes into distinct time windows, each associated with a different bunker price. By implementing this approach, the models presented in Chapters 5 and 6 could account for time-varying bunker prices. A similar methodology is employed in the study by Vilhelmsen et al. (2014). It is important to note that as the number of bunker nodes increases, solving the models becomes more challenging.

Speed Optimizations
In this thesis, it is assumed that vessels maintain a constant speed. However, vessels typically have different speeds when sailing in ballast or laden conditions. This distinction directly affects bunker consumption and travel costs. Therefore, it would make sense for a model to determine the optimal speed at which a vessel should travel. Speed optimization extensions have been extensively studied in previous research, including works by Norstad et al. (2011), Castillo-Villar et al. (2014), and Fan et al. (2019). Given the numerous studies on this topic, it would be reasonable to extend the models presented in this thesis with speed optimization techniques.

11.2 Solution Methods

All the suggested model extensions presented in Section 11.1 complicate the models and would increase the time complexity of finding an optimal solution. As such, the problem size of which the path flow solution method with a priori column generation is capable of solving is expected to decrease significantly. Thus, leveraging heuristic frameworks such as the presented ALNS-VCP matheuristic might be required. Future research leveraging such an ALNS-VCP matheuristic might incorporate a larger variety of Large Neighborhood Search (LNS) operators tailored to the specific problem at hand.

Other solution methods include dynamic column generation leveraged by Brønmo et al. (2010) and Vilhelmsen et al. (2014). In their studies, they formulate the path flow model with an associated subproblem which they solve as a Shortest Path Problem with Resource Constraints. Homsi et al. (2020) present an exact Branch-and-Price algorithm and a hybrid metaheuristic based on a Unified Hybrid Genetic Search.

11.3 Experiments

In this thesis, the impact of considering fleet repositioning was quantified by calculating the Value of the Stochastic Solution (VSS). Another approach would be to emplace the ALNS-VCP matheuristic in a simulation framework with sequential planning horizons. The number of vessels to allocate in each region represents a future outlook on the market and where cargoes should become available. The cargoes available in subsequent planning horizons could be generated according to distributions aligned to such outlooks. By simulating several sequential planning horizons, one could quantify the impact of considering fleet repositioning by comparing first-stage models and mean value models to the solutions of the stochastic two-stage formulation.

Bibliography

Aldworth, P., 2021. Lloyd's Maritime Atlas of World Ports and Shipping Places. Thirty-second ed., Informa Law from Routledge, London. doi:10.4324/9781003199885.

Andersson, H., Christiansen, M., Fagerholt, K., 2011. The Maritime Pickup and Delivery Problem with Time Windows and Split Loads. INFOR: Information Systems and Operational Research 49, 79–91. doi:10.3138/infor.49.2.079.

Archetti, C., Bertazzi, L., Hertz, A., Speranza, M.G., 2012. A Hybrid Heuristic for an Inventory Routing Problem. INFORMS Journal on Computing 24, 101–116. doi:10.1287/ijoc.1100.0439.

BalticExchange, 2022. History Timeline. https://balticexchange.com/en/who-we-are/history/baltic-timeline/intro.html. (accessed 2022-09-19).

Besbes, O., Savin, S., 2009. Going Bunkers: The Joint Route Selection and Refueling Problem. M&SOM 11, 694–711. doi:10.1287/msom.1080.0249.

Borthen, T., Loennechen, H., Wang, X., Fagerholt, K., Vidal, T., 2018. A genetic search-based heuristic for a fleet size and periodic routing problem with application to offshore supply planning. EURO Journal on Transportation and Logistics 7, 121–150. doi:10.1007/s13676-017-0111-x.

Brønmo, G., Christiansen, M., Fagerholt, K., Nygreen, B., 2007a. A multi-start local search heuristic for ship scheduling—a computational study. Computers & Operations Research 34, 900–917. doi:10.1016/j.cor.2005.05.017.

Brønmo, G., Christiansen, M., Nygreen, B., 2007b. Ship routing and scheduling with flexible cargo sizes. Journal of the Operational Research Society 58, 1167–1177. doi:10.1057/palgrave.jors.2602263.

Brønmo, G., Nygreen, B., Lysgaard, J., 2010. Column generation approaches to ship scheduling with flexible cargo sizes. European Journal of Operational Research 200, 139–150. doi:10.1016/j.ejor.2008.12.028.

Brown, G.G., Graves, G.W., Ronen, D., 1987. Scheduling Ocean Transportation of Crude Oil. Management Science 33, 335–346. doi:10.1287/mnsc.33.3.335.

Bulhões, T., Hà, M.H., Martinelli, R., Vidal, T., 2018. The vehicle routing problem with service level constraints. European Journal of Operational Research 265, 544–558. doi:10.1016/j.ejor.2017.08.027.

BunkerEx, 2022. BunkerEx — Bunker Prices — Bunker Broker United Kingdom. https://www.bunker-ex.com. (accessed 2022-11-11).

Castillo-Villar, K.K., González-Ramírez, R.G., Miranda González, P., Smith, N.R., 2014. A Heuristic Procedure for a Ship Routing and Scheduling Problem with Variable Speed and Discretized Time Windows. Mathematical Problems in Engineering 2014, e750232. doi:10.1155/2014/750232.

Christiansen, M., Fagerholt, K., 2014. Chapter 13: Ship Routing and Scheduling in Industrial and Tramp Shipping, in: Toth, P., Vigo, D. (Eds.), Vehicle Routing. Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 381–408. doi:10.1137/1.9781611973594.ch13.

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2007. Chapter 4 Maritime Transportation, in: Barnhart, C., Laporte, G. (Eds.), Handbooks in Operations Research and Management Science. Elsevier. volume 14 of *Transportation*, pp. 189–284. doi:10.1016/S0927-0507(06)14004-9.

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2013. Ship routing and scheduling in the new millennium. European Journal of Operational Research 228, 467–483. doi:10.1016/j.ejor.2012.12.002.

Christiansen, M., Fagerholt, K., Ronen, D., 2004. Ship Routing and Scheduling: Status and Perspectives. Transportation Science doi:10.1287/trsc.1030.0036.

Donovan, A., 2004. The Impact of Containerization: From Adam Smith to the 21st Century. Review of Business 25, 10–15.

dos Santos, P.T.G., Kretschmann, E., Borenstein, D., Guedes, P.C., 2020. Cargo routing and scheduling problem in deep-sea transportation: Case study from a fertilizer company. Computers & Operations Research 119, 104934. doi:10.1016/j.cor.2020.104934.

Elsevier, 2022. Scopus. https://www.elsevier.com/__data/assets/pdf_file/0017/114533/Scopus-fact-sheet-2022_WEB.pdf. (accessed 2022-11-20).

Erdoğan, S., Miller-Hooks, E., 2012. A Green Vehicle Routing Problem. Transportation Research Part E: Logistics and Transportation Review 48, 100–114. doi:10.1016/j.tre.2011.08.001.

Fagerholt, K., Christiansen, M., 2000a. A combined ship scheduling and allocation problem. Journal of the Operational Research Society 51, 834–842. doi:10.1057/palgrave.jors.2600973.

Fagerholt, K., Christiansen, M., 2000b. A travelling salesman problem with allocation, time window and precedence constraints — an application to ship scheduling. International Transactions in Operational Research 7, 231–244. doi:10.1111/j.1475-3995.2000.tb00196.x.

Fagerholt, K., Hvattum, L., Johnsen, T., Korsvik, J., 2013. Routing and scheduling in project shipping. Annals of Operations Research 207, 67–81. doi:10.1007/s10479-011-0888-1. cited By :11.

Fagerholt, K., Ronen, D., 2013. Bulk ship routing and scheduling: Solving practical problems may provide better results. Maritime Policy & Management 40, 48–64. doi:10.1080/03088839.2012.744481.

Fan, H., Yu, J., Liu, X., 2019. Tramp Ship Routing and Scheduling with Speed Optimization Considering Carbon Emissions. Sustainability 11, 6367. doi:10.3390/su11226367.

Fearnleys, 2022. Fearnleys Weekly Report. https://fearnpulse.com/. (accessed 2022-11-11).

Fischetti, M., Lodi, A., 2008. Repairing MIP infeasibility through local branching. Computers & Operations Research 35, 1436–1445. doi:10.1016/j.cor.2006.08.004.

Gatica, R.A., Miranda, P.A., 2011. Special Issue on Latin-American Research: A Time Based Discretization Approach for Ship Routing and Scheduling with Variable Speed. Netw Spat Econ

11, 465–485. doi:10.1007/s11067-010-9132-9.

Google, 2004. Google Scholar. https://scholar.google.com/. (accessed 2022-11-25).

Gurobi Optimization, LLC, 2022. Gurobi optimizer reference manual. https://www.gurobi.com. (accessed 2022-12-14).

He, L., Hu, Z., Zhang, M., 2020. Robust Repositioning for Vehicle Sharing. M&SOM 22, 241–256. doi:10.1287/msom.2018.0734.

Hemmati, A., Hvattum, L.M., Fagerholt, K., Norstad, I., 2014. Benchmark Suite for Industrial and Tramp Ship Routing and Scheduling Problems. INFOR: Information Systems and Operational Research 52, 28–38. doi:10.3138/infor.52.1.28.

Homsi, G., Martinelli, R., Vidal, T., Fagerholt, K., 2020. Industrial and tramp ship routing problems: Closing the gap for real-scale instances. European Journal of Operational Research 283, 972–990. doi:10.1016/j.ejor.2019.11.068.

Husby, E., 2022. Western Bulk, Internal Communication.

IMO, 2020. IMO 2020 - cleaner shipping for cleaner air. https://imopublicsite.azurewebsites.net/en/MediaCentre/PressBriefings/Pages/34-IMO-2020-sulphur-limit-.aspx. (accessed 2022-11-11).

Keskin, M., Çatay, B., Laporte, G., 2021. A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. Computers & Operations Research 125, 105060. doi:10.1016/j.cor.2020.105060.

Korsvik, J., Fagerholt, K., Laporte, G., 2010. A tabu search heuristic for ship routing and scheduling. Journal of the Operational Research Society 61, 594–603. doi:10.1057/jors.2008.192. cited By :47.

Korsvik, J.E., Fagerholt, K., 2010. A tabu search heuristic for ship routing and scheduling with flexible cargo quantities. J Heuristics 16, 117–137. doi:10.1007/s10732-008-9092-0.

Korsvik, J.E., Fagerholt, K., Laporte, G., 2011. A large neighbourhood search heuristic for ship routing and scheduling with split loads. Computers & Operations Research 38, 474–483. doi:10.1016/j.cor.2010.07.005.

Ksciuk, J., Kuhlemann, S., Tierney, K., Koberstein, A., 2022. Uncertainty in maritime ship routing and scheduling: A Literature review. European Journal of Operational Research , S037722172200649Xdoi:10.1016/j.ejor.2022.08.006.

Kuhlemann, S., Ksciuk, J., Tierney, K., Koberstein, A., 2021. The stochastic liner shipping fleet repositioning problem with uncertain container demands and travel times. EURO Journal on Transportation and Logistics 10, 100052. doi:10.1016/j.ejtl.2021.100052.

Lee, J., Kim, B.I., 2015. Industrial ship routing problem with split delivery and two types of vessels. Expert Systems with Applications 42, 9012–9023. doi:10.1016/j.eswa.2015.07.059.

Li, M., Fagerholt, K., Schütz, P., 2022. Stochastic tramp ship routing with speed optimization: Analyzing the impact of the Northern Sea Route on CO2 emissions. Ann Oper Res doi:10.1007/s10479-022-04923-w.

Lin, C., Choy, K.L., Ho, G.T.S., Chung, S.H., Lam, H.Y., 2014. Survey of Green Vehicle Routing Problem: Past and future trends. Expert Systems with Applications 41, 1118–1138. doi:10.1016/j.eswa.2013.07.107.

MaritimeOptima, 2022. Maritime Optima: Maritime Intelligence at your fingertips. https://www.maritimeoptima.com/. (accessed 2022-11-11).

Meng, Q., Wang, S., Lee, C.Y., 2015. A tailored branch-and-price approach for a joint tramp ship routing and bunkering problem. Transportation Research Part B: Methodological 72, 1–19. doi:10.1016/j.trb.2014.11.008.

Norstad, I., Fagerholt, K., Laporte, G., 2011. Tramp ship routing and scheduling with speed optimization. Transportation Research Part C: Emerging Technologies 19, 853–865. doi:10.1016/j.trc.2010.05.001.

Omholt-Jensen, S., 2022. Tramp ship routing and scheduling with bunker optimization in the dry bulk industry. Specialization Project, TIØ4500, Norwegian University of Science and Technology. Supervised by Kjetil Fagerholt and Frank Meisel.

Pache, H., Kastner, M., Jahn, C., 2019. Current state and trends in tramp ship routing and scheduling doi:10.15480/882.2504.

Rodrigue, J.P., 2020. The Geography of Transport Systems. Fifth edition ed., Routledge, Abingdon, Oxon ; New York, NY.

Ronen, D., 1983. Cargo ships routing and scheduling: Survey of models and problems. European Journal of Operational Research 12, 119–126. doi:10.1016/0377-2217(83)90215-1.

Ronen, D., 1993. Ship scheduling: The last decade. European Journal of Operational Research 71, 325–333. doi:10.1016/0377-2217(93)90343-L.

Ropke, S., Pisinger, D., 2006. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. Transportation Science 40, 455–472. doi:10.1287/trsc.1050.0135.

Rothberg, E., 2007. An Evolutionary Algorithm for Polishing Mixed Integer Programming Solutions. INFORMS Journal on Computing 19, 534–541. doi:10.1287/ijoc.1060.0189.

Shaw, P., 1998. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems, in: Maher, M., Puget, J.F. (Eds.), Principles and Practice of Constraint Programming — CP98, Springer, Berlin, Heidelberg. pp. 417–431. doi:10.1007/3-540-49481-2_30.

Stålhane, M., Andersson, H., Christiansen, M., 2015. A branch-and-price method for a ship routing and scheduling problem with cargo coupling and synchronization constraints. EURO Journal on Transportation and Logistics 4, 421–443. doi:10.1007/s13676-014-0061-5.

Stålhane, M., Andersson, H., Christiansen, M., Cordeau, J.F., Desaulniers, G., 2012. A branch-price-and-cut method for a ship routing and scheduling problem with split loads. Computers & Operations Research 39, 3361–3375. doi:10.1016/j.cor.2012.04.021.

Statista, 2022. Active shipyards worldwide 2014-2021. https://www.statista.com/statistics/1102673/active-shipyards-worldwide/. (accessed 2022-08-21).

Stopford, M., 2009. Maritime Economics. 3rd ed ed., Routledge, London ; New York.

Taillard, É., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.Y., 1997. A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. Transportation Science 31, 170–186. doi:10.1287/trsc.31.2.170.

Ulsrud, K.P., Vandvik, A.H., Ormevik, A.B., Fagerholt, K., Meisel, F., 2022. A time-dependent vessel routing problem with speed optimization. European Journal of Operational Research 303, 891–907. doi:10.1016/j.ejor.2022.03.015.

UNCOMTRADE, 2022. International Merchandise Trade Statistics. https://comtrade.un.org/. (accessed 2022-09-01).

UNCTAD, 2022. REVIEW OF MARITIME TRANSPORT 2021. UNITED NATIONS, S.l.

Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N., Rei, W., 2012. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. Operations Research 60, 611–624. doi:10.1287/opre.1120.1048.

Vilhelmsen, C., Lusby, R., Larsen, J., 2014. Tramp ship routing and scheduling with integrated bunker optimization. EURO Journal on Transportation and Logistics 3, 143–175. doi:10.1007/s13676-013-0039-8.

Wang, X., Norstad, I., Fagerholt, K., Christiansen, M., 2019. Green Tramp Shipping Routing and Scheduling: Effects of Market-Based Measures on CO2 Reduction, in: Psaraftis, H.N. (Ed.), Sustainable Shipping: A Cross-Disciplinary View. Springer International Publishing, Cham, pp. 285–305. doi:10.1007/978-3-030-04330-8_8.

Western Bulk, AS., 2022. Annual Report 2021. Technical Report.

Wilhelmsen, 2022. Suez Toll Calculator. https://www.wilhelmsen.com/tollcalculators/suez-toll-calculator/. (accessed 2022-11-11).

Wilson, R., 2013. The Principles of the Dry Bulk FFA Market. Research Collection Lee Kong Chian School Of Business. , 113.

NTNU

Norwegian University of
Science and Technology