Master's thesis

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

Fattir Ahmed Chaudhary
Håkon Stubhaug Drangsland
Erlend Johnsen Fossum

A Decomposition-Based Matheuristic for the Two-Echelon Multi-Product Maritime Inventory Routing Problem

Master's thesis in Industrial Economics and Technology Management
Supervisor: Magnus Stålhane
June 2022

NTNU
Norwegian University of
Science and Technology

Fattir Ahmed Chaudhary
Håkon Stubhaug Drangsland
Erlend Johnsen Fossum

A Decomposition-Based Matheuristic for the Two-Echelon Multi-Product Maritime Inventory Routing Problem

Master's thesis in Industrial Economics and Technology Management
Supervisor: Magnus Stålhane
June 2022

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

Preface

The purpose of this master's thesis is to present mathematical models and a matheuristic for solving two-echelon multi-product maritime inventory routing problems. The models formulated in this thesis are based on research that is already done within the field of maritime inventory routing problems, and are extended to suit problems with two echelons.

This master's thesis is written by three students from Industrial Economy and Technology Management at the Norwegian University of Science and Technology (NTNU), and concludes their Master of Science degree at NTNU. The master's thesis is a continuation of the project report carried out by Chaudhary et al. (2021), in the subject TIØ4500 - Managerial Economics and Operations Research.

We would like to thank our supervisor, Professor Magnus Stålhane, for his guidance, ideas, and educational discussions throughout the writing of this master's thesis as well as the project report.

Trondheim, June 2, 2022.

Summary

Maritime transport is the backbone of the global trade economy with more than 80% of the volume of international trade transported by sea. Being a slow mode of transportation, comprehensive scheduling is essential, and using mathematical optimization as a support tool can help reduce the costs of operating. Even though maritime inventory routing problems have been subject to a significant amount of research for the last decades, problems with two echelons and multiple products are not yet found in the literature.

In the problem at hand, multiple products are transported in a two-echelon supply chain. In the first echelon, the products are transported over long distances from a production port to regional hubs. Products are further distributed over shorter distances from the hubs to consumption ports in the second echelon. There is one region containing only the production port, with the remaining regions containing one hub and multiple consumption ports. Each region has a heterogeneous fleet of vessels. The objective is minimizing costs by deciding when, where, and how much of each product to transport, as well as how much and when to buy or sell in an external spot market available to all hubs.

Two arc-flow models for the problem are formulated, including a fixed-charge network flow formulation (FCNF), to which valid inequalities and variable bound tightenings are implemented. A decomposition matheuristic is developed with the aim of obtaining high quality solutions for larger instances. The matheuristic takes advantage of the problem structure by aggregating the demand in each region and splitting the problem into smaller subproblems, which are then solved iteratively. It applies the FCNF model, as this proved to be the best exact model for quickly finding high-quality solutions for the subproblems. Preprocessing techniques are applied, as well as a clustering technique combined with a relax-and-fix and fix-and-optimize framework when subproblems are too computationally heavy for the exact FCNF. In the computational study, the matheuristic finds feasible solutions to all 75 test instances, whereas the exact solution method finds feasible solutions to 37. Out of these 37 instances, the matheuristic finds the best objective value in 10, and the average objective value of the matheuristic is 0,9% higher than for the exact method. Analysis shows that the matheuristic on average removes around 70% of integer variables compared to the exact solution method.

Sammendrag

Sjøfart som transportmetode har lenge vært ryggraden i verdensøkonomien, og transporterer i dag omtrent 80% av den globale handelen per volum. Ettersom dette er en langsom transportsmetode, er det nødvendig med omfattende ruteplanlegging. Med bruk av matematisk optimering som beslutningsstøtte er det mulig å redusere operasjonskostnadene betydelig. Selv om maritime ruteplanleggingsproblemer har vært forsket mye på de siste tiårene, er det svært få studier som omfatter problemer med transport av flere produkter og flere ledd mellom produsent og forbruker.

Problemet i denne masteroppgaven omhandler transport av flere produkter i en verdikjede med mellomlagre mellom produsent og forbruker. I første ledd blir produkter transportert over lengre distanser fra en produsenthavn til de regionale mellomlagrene. Fra de regionale mellomlagrene blir produktene transportert videre over kortere distanser til forbrukere, og dette utgjør andre ledd. Det er én region som kun består av produsenten. De resterende regionene har ett regionalt mellomlager, og flere forbrukere. Hver region har en heterogen flåte med skip. Målet er å minimere kostnadene ved å avgjøre når, hvor, og hvor mye av hvert produkt som skal transporteres. I tillegg avgjøres det når og hvor mye som skal kjøpes og selges på spotmarkedet som er tilgjengelig for de reginalene mellomlagrene.

To matematiske modeller er presentert, inkludert en fixed-charge network flow (FCNF) med tilhørende gyldige ulikheter og innstramninger av variabelgrenser. En dekomponeringsheuristikk er implementert med mål om å finne løsninger av høy kvalitet på større instanser. Heuristikken utnytter problemstrukturen ved å aggregere etterspørselen i hver region, og deler problemet opp i subproblemer som løses iterativt. Den benytter FCNF-modellen som viste seg best egnet for å raskt finne løsninger for subproblemene. Preprosessering er benyttet, samt en kombinasjon av clustering og et relax-and-fix og fix-and-optimize rammeverk som benyttes når subproblemene blir for komplekse for den eksakte FCNF-modellen. I beregningsstuidet klarer heuristikken å finne løsninger for alle 75 instanser og den eksakte modellen finner løsinger for 37. Blant disse 37 instansene finner heuristikken beste objektivverdi i 10 av dem, og den gjennomsnittlige objektivverdien er 0,9% høyere enn for den eksakte modellen. Analyser viser at heuristikken i gjennomsnitt fjerner ca. 70% av heltallsvariablene sammenlignet med den eksakte løsningsmetoden.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the world has become more globalized, the production of different goods are specialized in certain geographic regions. At the same time, consumption of goods is increasing, which is met by increased levels of output. Reduction in trade barriers, and lower costs for raw materials and labour in certain areas of the world motivate companies to outsource to a greater extent. Consequently, the need for transporting products over longer distances is increasing. During the last 30 years, the international seaborne trade has increased by over 200% in terms of weight. In 2019, the seaborne trade passed 11 billion tons, which makes up closer to 90% of the total world trade (Sirimanne et al., 2019). Transport by sea has become the primary means of transporting several different products, including raw materials, crude oil, and petroleum products (Sirimanne et al., 2019). The importance of maritime transport in the global economy was emphasized by the previous Secretary-General of UN, Ban Ki-moon, when he referred to maritime transport as the "backbone of global trade and the global economy" (United Nations, 2016).

Due to the growth seen in maritime transportation in recent years, the number of ships in operation, and their sizes, have been increasing (Allianz, 2021). Operating large cargo ships with load capacities of several thousand tons is expensive. The main cost drivers in this industry include fuel costs, crew salaries and berthing fees. Additionally, there are large costs associated with either owning or chartering a vessel, where the latter cost is is primarily determined by the number of days a vessel is chartered. Maritime transportation is substantially slower than transportation by air or road, frequently taking several days, or even weeks, per transportation leg. Because many businesses rely on the goods being transported by sea, minor delays could negatively impact their entire operation. Hence, comprehensive planning concerning routing and inventory is important for keeping costs at a minimum. Using mathematical optimization as a support tool to create routes and schedules may potentially reduce the total costs of operation significantly. Another important aspect that favours optimizing the transportation routes by sea is the air pollution and emission of greenhouse gases caused by large cargo ships (EEA, 2021). These aspects are some of the reasons why maritime transportation has grown as a research field during this millennium.

In recent decades, optimizing maritime transport has become a greater research field of interest. The literature focuses on the problems concerning both the routing of the ships, as well as the inventory at the supplier and the customers. Supply chains implementing vendor-managed inventories are usually the subject for research. The problems within this research field are named *maritime inventory routing problems* (MIRP). Several case studies on the liquefied natural gas (LNG) industry were conducted throughout the first decade of this millennium. In the last decade, alongside several case studies, general models and extensions for MIRP have been published. Being a complex problem to solve, several different solution methods have also been developed to obtain good solutions within a reasonable time.

Existing MIRP literature primarily studies supply chains with no intermediate facilities between producer and consumer, and general mathematical models only represent this one-echelon structure. In reality, intermediate facilities are found in many supply chains. We study a general two-echelon multi-product maritime supply chain. In the first echelon, products are transported over longer distances from a production port to regional hubs. These hubs can also trade with an external spot market. In each region, there are multiple consumption ports served by their respective hubs. This constitutes the second echelon. The main costs considered are the fuel cost and the cost of buying and selling products in the spot market. When formulating this as a mathematical optimization problem, the objective is to minimize the total costs.

This masters' thesis contributes to the literature by studying a two-echelon multi-product MIRP, which is an extended version of the general MIRP. Two arc-flow formulations with these extensions are presented, with one being a basic arc-flow model, and the other a fixed-charge network flow model. The latter has been seen in the literature to achieve tighter bounds than the general model. To solve the larger instances, a matheuristic is presented. This heuristic is based on aggregating the customers' demand, and solves each region independently by applying clustering and using a relax-and-fix and fix-and-optimize framework.

This masters' thesis is organized in the following manner: Chapter 2 presents an overview of existing relevant literature. In Chapter 3, the problem studied is presented, and the two different mathematical formulations of the problem are given in Chapter 4. Chapter 5 presents the matheuristic for the formulated problem, and a computational study is conducted in Chapter 6. Concluding remarks are given in Chapter 7, and fields of study for future research are discussed in Chapter 8.

Chapter 2

Literature Review

In this chapter, relevant literature for maritime inventory routing problems is reviewed. In Section 2.1, the strategy for finding relevant literature is described. The general maritime inventory routing problem is discussed in Section 2.2, followed by model characteristics in Section 2.3. Extensions of the problem are discussed in Section 2.4 and Section 2.5, while Section 2.6 looks at solution methods. This review is based on work done in Chaudhary et al. (2021). Concluding the literature review with Section 2.7, we will discuss where the problem of this thesis situates in the existing research, as well as our contribution to the field.

2.1 Literature Search Strategy

The *maritime inventory routing problem*, hereby also known as *MIRP*, has developed a lot as a research field in recent years. Therefore, an increasing amount of literature is found concerned with problems within this field. It is mostly referred to as MIRP, but other abbreviations and names are also used for the same, or quite similar, problems in some papers.

To get an overview of the research done within this field, we started by looking at two surveys: Papageorgiou et al. (2014a) and Christiansen et al. (2013). These articles provided us with a good overview of the work done concerning MIRP, as well as working as a guideline for further literature search.

To begin with, the literature research was based on work referenced in the two surveys. As we started to narrow our search down towards specific extensions of the problem and solution methods, we used Google Scholar to search for relevant articles. We decided to only use articles that were written in English and that were published by acknowledged scientific publishers or in journals. Different combinations of words such as “two-echelon” and “multi-product” were used to retrieve relevant literature. An overview of the terms and search words used is shown in Table 2.1. The words written in the columns are words used in combinations with the headings when searching. Combinations of the keywords in each column have also been used.

Table 2.1: Words used in the search for relevant literature.

MIRP	**Multi-Product**	**Two-Echelon**	**MIRP Heuristic**
Mathematical Model	MIRP	MIRP	Relax-and-Fix
Maritime Inventory Routing Problem	IRP	IRP	Rolling Horizon
FCNF	Maritime Inventory Routing	Maritime Inventory Routing	Clustering
Arc-Flow	Inventory Routing	Inventory Routing	Fix-and-Optimize
	Unmixable	Heuristic	Decomposition
			Matheuristic

2.2 The General Maritime Inventory Routing Problem

Vendor managed inventory (VMI) is a supply chain practice where the supplier is in control of inventory management for the customers. In the literature, the *inventory routing problem* (IRP) deals with optimizing decisions in supply chains implementing VMI. It includes both inventory management and vehicle routing. In a general IRP, a single product is distributed from a single supplier to a set of customers to satisfy their demand for a given period. The consumer has a given consumption rate. Inventory capacities at the supplier and customers are given. The products are transported by a fleet of vehicles, which is usually homogeneous and with given capacities. The objective is to minimize the delivery and storage costs (Bertazzi and Speranza, 2012). A maritime inventory routing problem (MIRP) can be seen as a special case of the IRP that arises in a maritime setting (Papageorgiou et al., 2014a).

A MIRP is defined as *a planning problem where an actor has the responsibility for both the inventory management at one or both ends of the maritime transportation legs and for the ships' routing and scheduling* (Christiansen et al., 2013, p. 475). The goal of the general MIRP is to minimize the cost of transporting a single product between production and consumption ports. For both types of ports, there are given storage capacities, as well as known production and consumption rates. Usually, a heterogeneous fleet of ships is used for the transportation of the product. With a heterogeneous fleet, each ship has a distinct capacity. The ships can both load and discharge multiple times, often also partially, during the planning horizon. The initial inventories for ports and ships are known, as well as the initial locations of the ships. The costs of sailing, port costs, and waiting costs are ship-dependent. The objective is to minimize the costs by designing routes and schedules that minimize transport and delivery costs, as well as determining how much and at which ports to load/discharge the product (Christiansen et al., 2013).

2.3 Model Characteristics

The formulation of time is an important characteristic that differentiates mathematical models in MIRP from each other. The most common approach is to use discrete time. This is the approach Papageorgiou et al. (2014a) take in the first publicly available library of MIRP, where they present a core model for MIRP. In such models, each action occurs at a fixed point in time. Continuous time models are more often used in situations where production and consumption rates are constant, such as in the case of Al-Khayyal and Hwang (2007), where they minimize the cost of transporting several liquid bulk products. Another example of the use of continuous time is seen in the paper by Christiansen and Fagerholt (2009), where a single product, with constant production and consumption rates is transported from production to consumption ports.

Another important concept in the mathematical model is the difference between arc-flow and path-flow formulations. Models using path-flow formulation have decision variables that represent a whole sequence of ports that are visited by each ship. Persson and Göthe-Lundgren (2005) use this formulation when they look at the transportation of multiple oil products. Andersson (2011) takes the same approach when studying transportation of pulp products between mills and terminals. Arc-flow formulations have decision variables for movement from one port to another, and are more common in the MIRP literature than path-flow (Papageorgiou et al., 2014a). In the survey by Christiansen et al. (2013), Agra et al. (2013), and several other papers, arc-flow formulations are used.

Using an arc-flow and a discrete time-model to find optimal schedules for transporting a single product and maintaining capacities and restrictions in ports, Song and Furman (2013) present a time-space network. The problem is defined on a graph $G = (V, E)$, where V is the set of nodes that represent a visit to a specific port in a particular time period. The set of nodes includes both supplier and customers. E is the set of arcs that connects the nodes, and the arcs represent movement from one port to another in a later time period, or that the ship stays in the same port in subsequent time periods. The nodes are shared by all the ships, but each ship has its own set of arcs. This approach is also used by Persson and Göthe-Lundgren (2005) and Papageorgiou et al. (2014a). A visual representation of such a network for a single ship, with two ports and six time periods, is given in Figure 2.1. The source and sink nodes represent the start and end destinations for the ships.

With the goal of obtaining a better linear relaxation, Agra et al. (2013) extends this arc-flow model and formulate the MIRP as a single-commodity fixed-charge network flow problem (FCNF). A similar model is also used in Friske et al. (2021). In this formulation, the commodities supplied from the producer to the customers flow along the arcs corresponding to the vessel routes. The nodes in each port are divided into

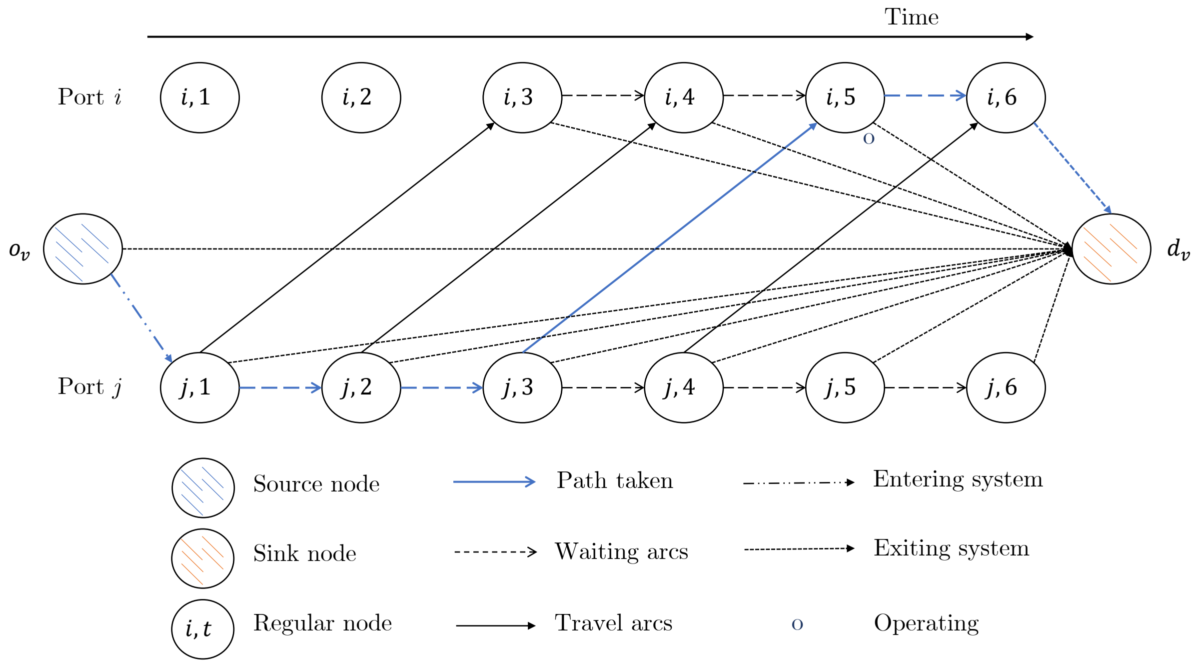

Figure 2.1: Visualization of time-space network.

three levels. The first level handles arrival and waiting of the vessels, the operations are controlled by the second level, and inventory is regulated in the last level. A visualization of this is given in Figure 2.2. A vessel arrives the port in first time time period and waits one day. Thereafter, it operates in two consecutive days before it leaves the port. With this reformulation, known valid inequalities can be applied to tighten the problem formulation, which makes it possible to find solutions more efficiently.

The composition of ships and their attributes in a fleet is another essential part that distinguishes the problems studied. A great majority of the aforementioned articles presents problems with heterogeneous fleets consisting of ships with different characteristics such as speed, capacity, and travel cost. In contrast, Andersson (2011) presents a problem where the fleet consists of three identical ships. The option of chartering ships is also present, as it is in Soroush and Al-Yakoob (2018) when transporting crude oil products. Stålhane et al. (2012) present a problem with an available fleet, in addition to the option to charter ships on daily rates when transporting multiple LNG products. When studying a problem of transporting products of grain, Bilgen and Ozkarahan (2007) do not have their own fleet, but charter the ships. Whether a ship can load or discharge in several ports also differ in the literature. As in Bilgen and Ozkarahan (2007), Soroush and Al-Yakoob (2018) can only discharge their products at a single port. In several other articles, such as Christiansen and Grønhaug (2009), Hemmati et al. (2016), and Hennig et al. (2012), the ships are allowed to load and discharge at several ports.

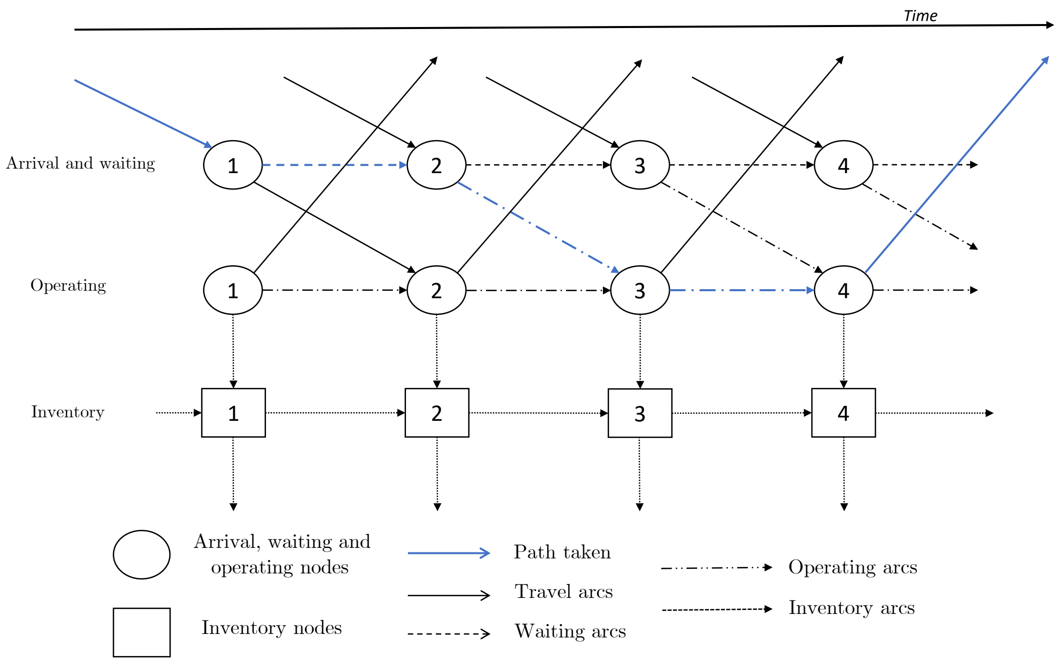

Figure 2.2: Visualization of the FCNF network.

2.4 Multi-Product MIRP Characteristics

The extension of multiple products is often seen in maritime inventory routing problems. Instead of one product, multiple products are to be transported and delivered. Depending on the problem, different parameters can be specified by product type, such as port demand and port inventory capacity. In some cases, it is not possible to mix the product types due to the nature of the products. In this section, the key characteristics of multi-product MIRP are discussed, which include bounds on loading, discharging and inventory at ports, as well as the inventory and product handling on board the ships.

Characteristics regarding loading, discharging and storing of the different products in the ports differ greatly in the literature. Papageorgiou et al. (2014a) introduce a general extension for multiple products in their core model. In their extension, there is no need for product-specific inventory, loading or discharging bounds on the ports, as the products are mixable. These bounds are set on the aggregate level. However, lower and upper bounds for the amount produced or consumed in each port are specified by product type. This extension stands out compared other papers, as unmixable products appear to be more common than mixable products in multi-product MIRP papers. A common approach is to set product-specific bounds for loading, discharging and inventory in ports. This approach is seen in the paper by Al-Khayyal and Hwang (2007), which describes such a problem with transportation of oil. Both upper and lower inventory bounds for each product in a port are implemented. The same is also done in Christiansen et al. (2011), where they look at transport of cement. This is directly transferable to the loading and discharging as well, since the storage capacity dictates these bounds. A different approach is taken by Hemmati et al. (2016), with all ports allowed to both produce and consume depending on the product. Each port is either producer, consumer or neither for a given product. Agra et al. (2014) present a combination of these approaches, with some ports able to both produce and consume products, and other ports only able to consume.

Proceeding with loading and discharging in ports, several different approaches for how ships are allowed to load and discharge can be found in the literature. Agra et al. (2014), Hemmati et al. (2016) and Siswanto et al. (2011) have taken the approach of restricting the ports to only have one ship visiting, and thus loading or discharging, at a given time. A similar problem is described by Christiansen et al. (2011). However, in this problem, only the consumption ports are restricted to one ship at a time. The production ports can service multiple ships simultaneously, as they are larger than the consumption ports. A less restrictive approach is taken by Al-Khayyal and Hwang (2007) by allowing only one ship to load or discharge a specified product in a given port at a time. Multiple ships can therefore load or discharge simultaneously as long as they handle different products. The most liberal approach is seen in the papers by Bilgen and Ozkarahan (2007) and Papageorgiou et al. (2014a), which both allow ships to simultaneously load and discharge multiple products.

A common challenge found in multi-product MIRP is the transportation of the products, and potentially the allocation of compartments on the ships. Unmixable products create a need for multiple compartments to separate the products. Bilgen and Ozkarahan (2007) deal with unmixable products, but take a similar approach as Papageorgiou et al. (2014a), where there is no need for tank allocation. They assume that there are enough compartments with varying sizes to maintain a separation of all products. Therefore, as in Papageorgiou et al. (2014a), the total capacity of the ship is the only capacity constraint to consider for the ships. A similar approach is seen in the papers by Hemmati et al. (2016) and Persson and Göthe-Lundgren (2005). Siswanto et al. (2011) introduce compartments for unmixable products, but use undedicated ones. Undedicated compartments are defined as compartments that can transport any type of product, but only one type at a time. In contrast to the previous papers mentioned, the number of compartments is not high enough to assume that separation of the products is possible with all combinations of quantities. Christiansen et al. (2011) take a similar approach, with undedicated compartments for the unmixable cement products. Agra et al. (2014) and Al-Khayyal and Hwang (2007) both introduce dedicated compartments. Consequently, they need to address the problem of allocating products to different compartments, adding an allocation problem to their MIRP. Dedicated compartments imply that a given compartment transports one type of product during the entire planning period.

2.5 Two-Echelon Inventory Routing Problems

Standard routing problems consider supply chains of only two layers. Products move directly from origin to destination, which is referred to as direct shipping. When supply chains use one or more layers of intermediate facilities for storage, merging, consolidation or transsshipment, it is defined as a multi-echelon. Two-echelon is a special case of multi-echelon with one layer of intermediate facilities. The problem can not be decomposed and solved separately for each echelon since the flow of freight must be coordinated between layers in the supply chain (Cuda et al., 2015). When conducting the literature review, it appeared that no, or very few, papers study multi-echelon MIRPs. Consequently, this section reviews relevant literature on the closely related two-echelon IRP, which deals with both vehicle routing as well as inventory management on a supply chain of two echelons.

Although there are many similarities between MIRP and IRP, a few key distinctions are prominent. Some of those Ronen (2002) outlines are: (1) In IRP, the vehicles in a fleet are usually identical. This is usually not the case in MIRPs, where fleets are often heterogeneous. (2) MIRPs often involve several products, and sometimes products that have to be kept separately, which is rarely necessary in IRP. (3) Travel time is often much longer in MIRP, and (4) unlike the ships, the vehicles in IRP must return to base each day. (5) Ships can also often be sourced at several facilities in MIRPs, unlike the vehicles in IRPs.

In the two-echelon inventory routing problem (2E-IRP) literature, the number of intermediate facilities and origins differ. Chan and Simchi-Levi (1998) study a distribution system with one origin, multiple intermediate facilities, and multiple destinations where there is an upper limit to the frequency of visits. In Farias et al. (2021) a similar distribution system is studied, but without frequency restrictions. Here, one vehicle can supply multiple intermediate facilities before restocking at the origin. Both Li et al. (2011) and Zhao et al. (2008) study a distribution system with one intermediate facility and one origin, where the former allows direct shipment from origin to destination. Rohmer et al. (2019) introduce a two-echelon multi-product IRP for perishable goods with one origin and one intermediate facility. Guimarães et al. (2019) claim to be the first to introduce a formulation with multiple origins and intermediate facilities in their paper on distribution systems, which is based on an ethanol and gasoline supply chain in South America.

The modelling of the fleet of vehicles also differs in the literature. In Guimarães et al. (2019), each intermediate facility has one fleet of vehicles assigned to it. All fleets are homogeneous and equal. These vehicles are used for transportation in both echelons. Vehicles must return to their original facility before the end of the planning period. In contrast, Chan and Simchi-Levi (1998), Li et al. (2011), and Farias et al. (2021) have two homogeneous fleets; one for each echelon. Zhao et al. (2008) study a problem where goods in the first echelon are transported by train, while a homogeneous fleet of vehicles is used in the second echelon.

2.6 Solution Methods

The combination of vehicle routing and inventory management makes MIRPs difficult to solve using commercial solvers, such as CPLEX and Gurobi, especially for larger instances. Only a few papers apply exact solution methods to their problems, and will therefore be briefly reviewed in section 2.6.1. Due to the complexity of MIRPs, several different heuristic algorithms are found in the literature. Some papers only use metaheuristics to solve the problem, which are discussed in section 2.6.2, but the great majority of them are *matheuristics*. A matheuristic combines mathematical programming and metaheuristics, which are general heuristic frameworks for problems with similar patterns, to create an optimization algorithm (Fischetti and Fischetti, 2016). In the *vehicle routing problem* (VRP) literature, which is closely related to IRP, with the main difference being that the amount to transport is given in the VRP, Archetti and Speranza (2014) divides the matheuristics into three categories: (1) decomposition approaches, where the problems are divided into smaller subproblems, (2) improvement heuristics and (3) branch-and-price/column generation-based approaches. As MIRP is a special case of IRP, this division appears appropriate when studying the MIRP literature as well. Due to the differences in problem characteristics between IRP and MIRP, the main focus in this section is on solution methods in the literature applied on MIRPs. In this section, the first two categories, decomposition approaches and improvement heuristics, are reviewed in subsections 2.6.4-2.6.5, respectively. In combination with matheuristics, several papers also include preprocessing to make the problems easier to solve, which is discussed in Subsection 2.6.3.

2.6.1 Exact Solution Methods

One of the main purposes of solving the problems using exact solution methods is to prove, and solve problems to, optimality. To test the new FCNF formulation, and the valid inequalities, Agra et al. (2013) conducted a computational study using a branch-and-bound scheme. Different branching priorities were set on variables concerning the vessel movements. The results show that compared to the original formulation of the problem, the new FCNF formulation provides much better bounds. When studying the transportation of multiple oil-products from oil refineries to depots, Persson and Göthe-Lundgren (2005) use column generation and constraint branching as the solution method.

The shipment problem, concerning products and amounts to be transported, is linearly relaxed, and solved using column generation. To choose among the schedules obtained for the ships, constraint branching is used. The branching implies that a ship must visit a port in a given time period, or it can not. Valid inequalities are also applied to obtain solutions more quickly. A similar approach is used when Engineer et al. (2012) look at a single-product MIRP with multiple supply and demand facilities. A branch-and-price algorithm is used to solve the master problem containing all possible routes, in addition to load and discharge quantities. The linear relaxation of the master problem is solved by cutting the branch-and-bound tree if the solution is fractional. A similar branching scheme as Persson and Göthe-Lundgren (2005) is imposed, where a vessel is either forced to or prohibited from loading or discharging a product at a port at a given time. With these methods, larger instances of MIRPs can be solved. Studying transportation of LNG, Grønhaug et al. (2010) also use a branch-and-price method. The master problem in the column generation handles the inventory, and the subproblems generate the ship routes. The solution method finds high-quality integer solutions on almost all instances tested.

2.6.2 Metaheuristics

Metaheuristics are general algorithm ideas that can be applied on several different problems to obtain solutions more efficiently than exact methods, including MIRPs. When optimizing the solution of an ADP problem for LNG transportation, Stålhane et al. (2012) use a greedy construction algorithm that gives a set of initial solutions, combined with a first-descent neighborhood search with five neighborhood operators. High-quality solutions are obtained in a short amount of time compared to exact solution methods. Also studying LNG problems, Asokan (2014) proposes an algorithm with parallel large neighborhood search to solve larger instances. Initially, a construction heuristic is used to build feasible solutions, and thereafter improved by neighborhood search. The operators are either a pair of selected vessels, or time-windows. Based on these operators, new subproblems are created to optimize the schedules within the neighborhoods. Using this method, good results are achieved on the test instances within reasonable time. When studying a multi-product MIRP, Sanghikian et al. (2021) apply a variable neighborhood search heuristic. Four neighborhoods are defined, with the first two as intra-route, where for example two visits in the solution are exchanged. The two other are inter-route, meaning inserting or deleting a visit in a route. Compared to work done prior to this paper, this method performs better in terms of objective value and computational time.

2.6.3 Preprocessing

The purpose of preprocessing is to reduce the problem size, which is often done by enforcing problem-specific requirements. This includes both fixing and/or removing some of the variables. In their arc-flow model, Song and Furman (2013) remove several arcs between consumption and production ports, and set the ship storage to zero at the end of the planning horizon. A similar approach is taken by Friske et al. (2021), where they study the MIRP presented in Papageorgiou et al. (2014a). The problem is formulated on a time-space network, and arcs that have a low probability of being used in a high-quality solution are removed in preprocessing. Examples of such arcs are those that have very long travel distances. Reduction of arcs that ships can use is also found in Papageorgiou et al. (2018). When solving the problem, only arcs that facilitate travel in even-numbered periods are generated.

To reduce the solution space, Rakke et al. (2011) remove solutions that are almost symmetric. This is done by reducing the number of contracts a ship can deliver to on a given day. This improves both the solution time and the quality of solutions. Fagerholt and Christiansen (2000) also reduce the complexity of the problem when studying a ship scheduling problem by reducing the number of candidate schedules. Schedules with poor utilization of ships are most likely not included in an optimal solution and are therefore excluded.

2.6.4 Decomposition Approaches

In decomposition approaches, the main problem is divided into subproblems, and solved iteratively. As will be seen in this section, it is common to decompose with regard to time periods. In a general *rolling horizon heuristic* (RHH), the problem is divided into n time periods, giving n subproblems. Each subproblem is solved as a mixed integer problem, and by combining these solutions, a solution for the full problem is obtained. Studying a MIRP for an LNG producer, Rakke et al. (2011) use an RHH to create an annual delivery program (ADP). They divide the whole planning horizon into a number of time partitions, where some of the partitions might be overlapping, and solve each subproblem with branch-and-bound. This method provides high quality solutions, and solution times are significantly lower than when solving the full mathematical model using a commercial solver. Papageorgiou et al. (2018) present a computational study using RHH to find schedules and inventory policies for MIRPs, which solves several instances of different MIRPs significantly quicker than CPLEX and Gurobi. Looking at the optimization of short sea fuel oil distribution, Agra et al. (2014) use a hybrid heuristic consisting of three heuristics. An initial solution is found by the first heuristic, which is then solved by RHH, before local optima are found by using local branching. The use of the three heuristics combined outperformed the use of only the RHH.

Uggen et al. (2013) extend the RHH model and present a relax-and-fix (R&F) heuristic. The main difference from the RHH is that R&F considers the whole planning horizon in each iteration, but integer requirements on variables outside the subproblem are relaxed. When using RHH, only the time horizon that makes up one subproblem is considered in each iteration, and the remaining integer variables are ignored. R&F heuristics solves each subproblem with integer variables, while it relaxes the variables in the remaining subproblems that are not yet solved. Once the subproblem is solved, the variables in the solved subproblem are fixed. It iterates through every subproblem until all the variables in the problem are fixed. An illustration of the R&F heuristic is given in Figure 2.3.

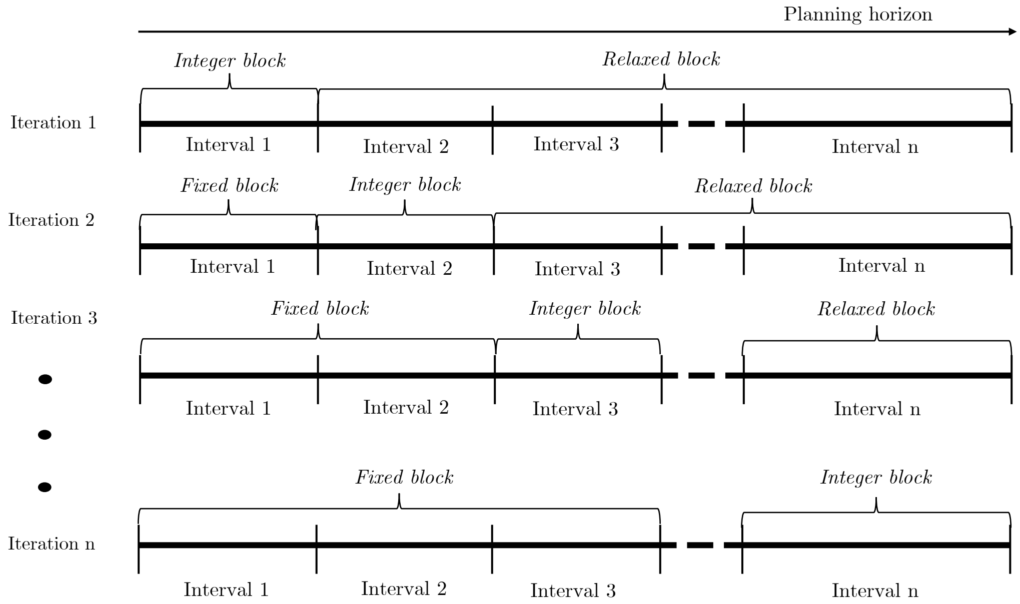

Figure 2.3: Visualization of Relax-and-Fix heuristic.

Building on the model presented in Uggen et al. (2013), Friske et al. (2021) present an effective R&F algorithm for the problem presented in Papageorgiou et al. (2014a). To avoid infeasibility, they also use overlapping planning horizons. The R&F heuristic is used to generate an initial solution, and to improve this solution they use a fix-and-optimize (F&O) heuristic. In the R&F, the problem is divided into a number of subproblems which are made up by time intervals in the planning horizon. In each iteration, the planning horizon is dived into three blocks. The first is an integer block where the integer requirements are forced. The second and third are a relaxed and an end block, where integer requirements are relaxed. After solving the first iteration, the time period that made up the integer block is now a fixed block, where the solution is fixed, and the proceeding blocks become an integer block, relaxed block and end block, respectively. This is done iteratively until the last time interval is reached. The proposed algorithm was implemented on both an arc-flow model and an FCNF formulation, based on the work of Agra et al. (2013). It was more efficient when used on the FCNF formulation, and it obtained three new best-known solutions on the public dataset presented in Papageorgiou et al. (2014a). Similar solution methods were implemented on a single-product MIRP in Friske and Buriol (2017), which used an arc-flow formulation, and in Friske and Buriol (2018), where an FCNF formulation was used. Both managed to obtain new best-known solutions for a few instances.

The partitioning of variables to create subproblems can also be done based on other criteria than time. Ferreira et al. (2009) show that one can use the different indices of the integer variables to group variables and create more tailor-made subproblems. Papageorgiou et al. (2014b) present a two-stage decomposition algorithm for a MIRP. In the first stage, which is the construction phase, ports are aggregated by regions and vessels by vessel classes. In the "zoomed out" phase, the master-problem concerning the routing between regions is solved by using the aggregated information. In the "zoomed in" phase, each region makes up a subproblem, in which the vessel routes are decided, and how much to load and discharge. In the second stage, the solution from the first stage is used, and valid inequalities and practical assumptions are imposed, such as limiting how many customers each vessel can visit. Computational results show that combining these two phases outperforms commercial MIP solvers. Another approach taking advantage of the problem structure is found in Hemmati et al. (2016), where they consider a multi-product MIRP and use a hybrid matheuristic. In the first phase, the inventory-routing problem is converted into a ship routing and scheduling problem. This is done by generating cargo subject to the inventory limits. In the second phase, the problems are solved by using neighborhood search. This approach is called Hybrid Cargo Generating and Routing (HCGR), and is built on the approach of Hemmati et al. (2015).

Guimarães et al. (2019) present a matheuristic for solving larger instances of the 2E-IRP by handling vehicle routes with an Adaptive Large Neighbourhood Search (ALNS), while a MIP is solved exactly to decide delivery quantities and the pickup of input from supplier. The MIP is also allowed to add and remove customers from a route to potentially find better neighbour solutions. Rohmer et al. (2019) present a matheuristic combining ALNS with a MILP formulation for the 2E-IRP. Three versions of the approach is presented, differing in the extent to which routing and delivery patterns are solved independently of each other. Vadseth et al. (2021) look at an IRP with a maximum level replenishment policy with the aim to minimize transportation and inventory costs. A matheuristic, which initially generates a giant tour and then splits it into routes to generate an initial set of promising routes, is presented. It iterates between solving a path-flow model with a small set of routes, and updating the set of routes based on the optimal solution from the previous iteration. This metaheuristic is very efficient on IRPs, and finds the best-known solutions for 179 out of 240 multi-vehicle IRP benchmark instances.

Another approach to decomposing problems seen in the IRP literature is to partition the set of customers into disjoint subsets called *clusters*. Studying the supply chain of a supermarket chain in the Netherlands, Gaur and Fisher (2004) use clustering of customers when developing weekly delivery schedules. They start by clustering two customers at a time, before using a heuristic that expands these clusters to obtain better solutions. Using this method, the supermarkets reduced their transport expenses significantly. Nambirajana et al. (2016) also take advantage of clustering when studying a 2E-IRP. In their three-phase heuristic to solve the problem, clustering is an essential phase. They present both clustering of customers based on distance to reduce the travel costs, and also based on the total demand and vehicle capacities. The clustering is an iterative process that is repeated to avoid unfavourable clusters. Using this solution method, they achieve better results than using the commercial solver CPLEX.

2.6.5 Improvement Heuristics

In some cases, matheuristics are used as improvement procedures. As mentioned, Friske et al. (2021) use an F&O heuristic to improve the initial solution provided by the R&F. In F&O, a set of integer values from the solution given by R&F are unfixed, and the full problem is optimized once again. By doing this iteratively for each subproblem, the problem is re-optimized to obtain better solutions. Song and Furman (2013) present a modeling framework for the IRP and look at a specific case study on a MIRP. The heuristic used is an optimization-based large neighborhood search. The search for a better solution starts with a feasible solution from a branch-and-cut algorithm. Some of the binary variables are fixed to the values from the current solution, and small subproblems are created and optimized. The subproblems are generated by fixing all binary variables except those associated with a selected pair of ships. The subproblems optimize the transportation cost for the two selected ships and fix the variables, before iterating to the next subproblem which includes a new pair of ships. Goel et al. (2012) present an arc-flow formulation based on the MIRP model from Song and Furman (2013). They look at optimizing shipping schedule and inventory management in an LNG project. A greedy construction algorithm is used to build an initial solution, which is improved by two large neighborhood search algorithms. The first algorithm changes the departure date of the ships, while the other improves the routes of two ships at a time. This algorithm works efficiently and is faster than using commercial solvers.

2.7 Our Contribution

Maritime inventory routing problems have been studied for many years and have grown as a research field during the last decade. Routing and scheduling for LNG companies and optimizing transport of different products, such as crude oil and pulp, are some examples of case studies categorized as MIRPs. For many years, most MIRPs were case-specific, but as MIRP has become a more common subject of research, a library for MIRP has been introduced by Papageorgiou et al. (2014a). In recent years, several general extensions of the MIRP have been presented.

The general mathematical formulation of the studied general two-echelon multi-product MIRP in this master's thesis is based on the core MIRP model presented by Papageorgiou et al. (2014a). Studying the results in Chaudhary et al. (2021), the need for a model with tighter dual bounds was prominent. Therefore, extending this model, based on the work of Agra et al. (2013) and Friske et al. (2021), an FCNF formulation of the problem at hand is also presented. The problem is defined on a time-space network. The decision variables regarding ship movement are connected to arcs, and similar arc-flow models are also used in Christiansen et al. (2013) and Agra et al. (2013). The fleet consists of heterogeneous ships, which is also the case for Al-Khayyal and Hwang (2007), Song and Furman (2013) and Christiansen and Fagerholt (2009). Unlike Goel et al. (2012) and Soroush and Al-Yakoob (2018), but alike Hemmati et al. (2016) and Al-Khayyal and Hwang (2007), the ships are allowed to partially load and/or discharge. The products in this problem are unmixable, but during transportation, they are treated similarly as in Bilgen and Ozkarahan (2007). It is assumed that there are enough compartments with varying sizes on the ships to maintain the separation of all products, and that they are only limited by the total storage capacity of the ship. This means that tank allocation is not taken into account in the formulation. Just like Papageorgiou et al. (2018) and Uggen et al. (2013) the option of buying and selling products from a spot market is available. The objective is to minimize the cost of delivering the products demanded.

As the instance sizes grow, MIRPs quickly become difficult to solve using exact methods and commercial solvers. In recent years, several heuristics approaches have been made to obtain high-quality solutions with shorter computational time. Concerning two-echelon, only heuristics solving IRPs are found in the literature. Due to the differences in the nature of IRPs and MIRPs, which are mentioned in Section 2.5, the existing solution methods are not easily applicable for the problem studied. Therefore, a new matheuristic is presented. Inspiration is initially taken from Song and Furman (2013) and Friske et al. (2021) to reduce the problem size by preprocessing. Certain arcs are removed and some variables are fixed. The matheuristic in this paper is further inspired by Papageorgiou et al. (2014b), which aggregate regions into super-customers. This creates subproblems that consists of single echelon MIRPs. In each region, a clustering of customers are made, similar to Gaur and Fisher (2004) and Nambirajana et al. (2016). A relax-and-fix and fix-and-optimize heuristic is applied to the clusters, which provided high-quality solutions in short computational time in Friske et al. (2021), among others. Utilizing the structure of the problem, the main problem is divided into subproblems, which are solved and improved iteratively to obtain high-quality solutions for the whole problem.

During the literature study, we have not come across any studies concerning multi-echelon MIRPs. Building on the work from Papageorgiou et al. (2014a) and Agra et al. (2013), we present a general arc-flow formulation, and a new FCNF formulation with tightenings and valid inequalities, for the two-echelon multi-product maritime inventory routing problem. We also present a new decompisition-based matheuristic with elements from clustering and relax-and-fix and fix-and-optimize heuristics, which demonstrates how the structure of the problem can be utilized. A computational study is conducted, where the performances of the two mathematical formulations are compared, as well an extensive testing of the presented solution method.

Chapter 3

Problem Description

In this master's thesis, a general two-echelon multi-product maritime inventory routing problem with heterogeneous fleets is studied. There is a set of unmixable products, which must be kept separately for the entire planning period. The set of ports contains one production port, several regional hubs, and multiple consumption ports.

All ports have a storage for each product with a given maximum and minimum capacity. Each port has an initial inventory of each product at the start of the planning period. To avoid greedy or short-term solutions, the quantity stored of each product in each port must be greater than or equal to the average of the maximum and minimum capacity at the end of the planning period. Furthermore, there is a berth capacity in each port. There are bounds on the quantity that can be loaded or discharged in a port by a vessel per time period. The rate of production in the production port and the demand in the consumption ports of each product are given for each time period. The hubs have neither production nor demand, but are connected to a spot market. All hubs can buy and sell an unlimited quantity of each product in all time periods at a fixed buying and selling price.

In Figure 3.1, an example of the ports in a problem instance of 4 regions is illustrated. There is one region containing only the production port, hereby referred to as the production region. The remaining regions, referred to as consumption regions, contain one hub and multiple consumption ports. There is one heterogeneous fleet of vessels supplying the hubs with products from the production port. These vessels will never travel directly from one hub to another, and will thus visit the production port between visits to hubs. No direct shipments from production to consumption ports are allowed. For each consumption region, there is a heterogeneous fleet of vessels supplying the consumption ports with products from the hub. Consumption ports are only supplied by the hub in the same region. A vessel may visit and discharge at multiple consumption ports before restocking at the hub. Vessels can not visit ports outside its region. If a vessel visits a port, it must operate for at least one time period and may continue to operate after this. Waiting can only happen prior to operating. There are no vessels in any of the ports at the end of the planning horizon. At the hubs, vessels from the production region and consumption region will only discharge and load, respectively. When visiting a production port, a vessel can only load. In consumption ports, discharging is the only option. The transportation of product to and from the spot market is assumed to be instant and handled externally.

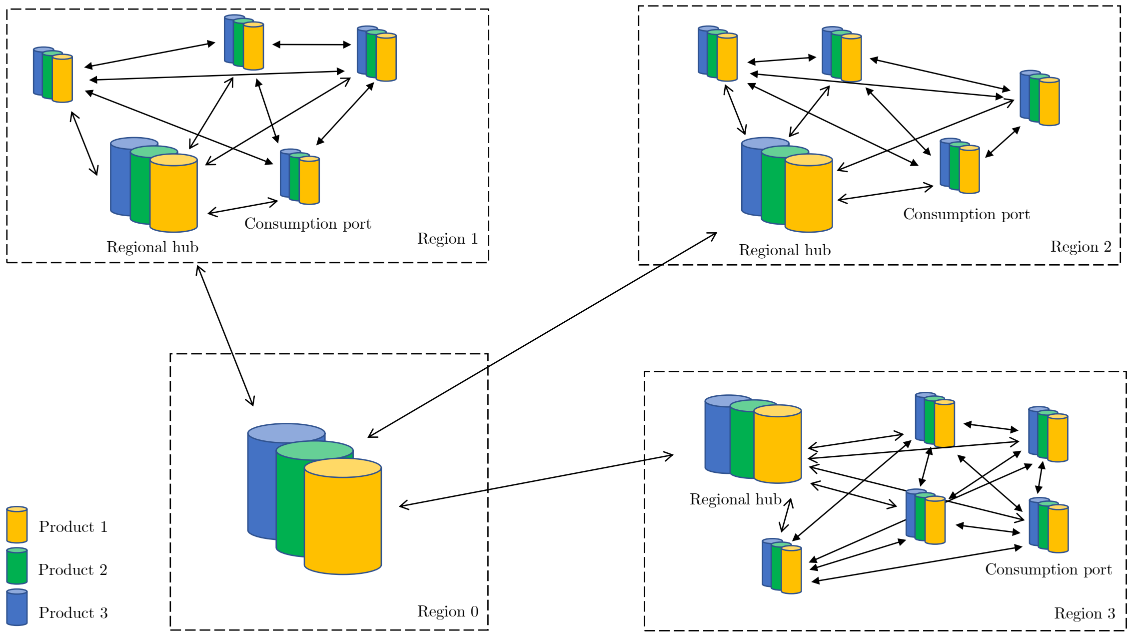

Figure 3.1: Example of ports in a problem instance with 4 regions and 3 products.

Each vessel has a given storage capacity for the total quantity of all products. There is a cost associated with choosing to travel from one port to another with a given vessel. A vessel has a given time period from which it is first available, and an initial inventory of each product at that time. A vessel can load or discharge multiple products simultaneously. Multiple vessels can load or discharge the same product at the same time in a port, within the bounds of the berth capacity.

The objective is to minimize the total costs, which consist of the travelling cost and cost of products bought and sold in the spot market. The decisions to be made are whether or not a vessel will sail from one port to another in a given time period, and how much to load or discharge of each product when it operates. Additionally, the quantity of a product bought or sold in the spot market from or to a hub at a given time must be decided.

Chapter 4

Mathematical Model

As discussed in Section 2.7, arc-flow discrete-time formulations can be used to represent the problem at hand. In this section, two different formulations will be presented. The first model, presented in Section 4.1, is based on the formulation presented by Papageorgiou et al. (2014a). The second model, presented in Section 4.2, takes inspiration from Agra et al. (2013) and Friske and Buriol (2018), and presents the problem as a fixed-charge network flow problem (FCNF). Inspired by the work of these two papers, we present known valid inequalities and variable tightenings for FCNF which are modified to fit the problem at hand.

4.1 Basic Arc-Flow Formulation

The model is based on a time-space network that consists of nodes, which represent the combination of a port and a specific time period. The nodes are connected with arcs. A vessel sailing from the source node to a regular node represents the arrival of the vessel to its initial destination. A vessel travelling on an arc from a regular node to the sink node indicates that the vessel is no longer being used. A vessel using an arc between two nodes associated with the same port, but at different times, indicates that the vessel is either waiting or operating in that port. A vessel sailing directly from the source node to the sink node is equivalent to not using the vessel. Arcs are permitted for a vessel if, and only if, the ports associated with the nodes belong to a region to which the vessel can travel. Additionally, permitted arcs must allow for sufficient sailing and loading time.

4.1.1 Modelling Assumptions

1. Vessels have numerous tanks, and it is assumed that combinations of tanks allow for every ratio of products as long as the total quantity is below the capacity of the vessel. Hence, the vessel capacity constraints only consider the total quantity of products stored.

2. A vessel can not start to operate at a port in the last period of the planning horizon.
3. A time-discrete model has been used. Variables are therefore only changed at separate points in time. A given point in time is always rounded to the nearest time period.

4.1.2 Definitions

Indices

t - time period
r - region
v - vessel
j - port
p - product type
n - node, $n = (j,t) : j \in \mathcal{J}, t \in \mathcal{T}$
a - arc, $a = ((i,t),(j,t')) : i,j \in \mathcal{J}, t,t' \in \mathcal{T}, t' > t$

Sets

$\mathcal{T}$ - set of time periods $\{0, |\mathcal{T}|\text{-}1\}$,
$\mathcal{R}$ - set of regions, where $r = 0$ is the production region and remaining regions are consumption regions
$\mathcal{V}$ - set of all vessels
$\mathcal{V}_r$ - set of vessels associated with region $r \in \mathcal{R}$
$\mathcal{J}^{\mathcal{P}}$ - set of production ports
$\mathcal{J}^{\mathcal{C}}$ - set of consumption ports
$\mathcal{J}^{\mathcal{H}}$ - set of regional hubs
$\mathcal{J}$ - set of all ports, $\mathcal{J} = \mathcal{J}^{\mathcal{P}} \cup \mathcal{J}^{\mathcal{C}} \cup \mathcal{J}^{\mathcal{H}}$
$\mathcal{J}_v$ - set of ports which vessel v can visit
$\mathcal{J}_r$ - set of ports in region r
$\mathcal{P}$ - set of product types
$\mathcal{N}$ - set of nodes, excluding source node n_0 and a sink node $n_{\mathcal{T}}$
$\mathcal{N}_0$ - set of all nodes, including a source node n_0 and a sink node $n_{\mathcal{T}}$
$\mathcal{N}^{\mathcal{P}}$ - set of nodes $n = (j,t)$ where $j \in \mathcal{J}^{\mathcal{P}}$
$\mathcal{N}^{\mathcal{C}}$ - set of nodes $n = (j,t)$ where $j \in \mathcal{J}^{\mathcal{C}}$
$\mathcal{N}^{\mathcal{H}}$ - set of nodes $n = (j,t)$ where $j \in \mathcal{J}^{\mathcal{H}}$
$\mathcal{A}$ - set of all arcs
$\mathcal{A}_v$ - set of (permitted) arcs associated with vessel $v \in \mathcal{V}$. This set contains only arcs leaving a node on, or later than, the first time period from which this vessel is available.
δ^+_{vn} - outgoing arcs associated with node $n = (j,t) \in \mathcal{N}_0$ and vessel $v \in \mathcal{V}$

δ^-_{vn} - incoming arcs associated with node $n = (j, t) \in \mathcal{N}_0$ and vessel $v \in \mathcal{V}$

Parameters

B_j - number of berths in port $j \in \mathcal{J}$
C_{va} - cost for vessel v to traverse arc $a \in \mathcal{A}_v$
C^B - cost per quantity bought on the sport market
C^S - revenue per quantity sold on the sport market
D_{jtp} - quantity produced or consumed of product $p \in \mathcal{P}$ at port $j \in \mathcal{J}$ in period $t \in \mathcal{T}$. The parameter is negative for $j \in \mathcal{J}^{\mathcal{C}}$, and positive for $j \in \mathcal{J}^{\mathcal{P}}$.
Δ_j - an indicator parameter taking value $+1$ if $j \in \mathcal{J}^{\mathcal{P}}$ and -1 if $j \in \mathcal{J}^{\mathcal{C}}$
F^{min}_j - minimum quantity that can be loaded or discharged to or from a vessel at port $j \in \mathcal{J}$ in each time period
F^{max}_j - maximum quantity that can be loaded or discharged to or from a vessel at port $j \in \mathcal{J}$ in each time period
Q_v - storage capacity of vessel $v \in \mathcal{V}$
S^{max}_{jp} - upper bound on inventory for product $p \in \mathcal{P}$ at port $j \in \mathcal{J}$
S^{min}_{jp} - lower bound on inventory for product $p \in \mathcal{P}$ at port $j \in \mathcal{J}$
S^0_{jp} - initial storage of product $p \in \mathcal{P}$ at port $j \in \mathcal{J}$
U^0_{vp} - initial storage of product $p \in \mathcal{P}$ in vessel $v \in \mathcal{V}$

Variables

f_{jvtp} - quantity loaded or discharged of product $p \in \mathcal{P}$ at port $j \in \mathcal{J}$ in period $t \in \mathcal{T}$ for vessel $v \in \mathcal{V}$
s_{jtp} - quantity of product $p \in \mathcal{P}$ at port $j \in \mathcal{J}$ available at the end of period $t \in \mathcal{T}$
u_{vtp} - quantity of product $p \in \mathcal{P}$ on vessel $v \in \mathcal{V}$ available at the end of period $t \in \mathcal{T}$
x_{va} - takes value 1 if vessel $v \in \mathcal{V}$ uses arc $a \in \mathcal{A}_v$, and 0 otherwise
z_{jvt} - takes value 1 if vessel $v \in \mathcal{V}$ loads or discharges at node $n = (j, t) \in \mathcal{N}$, and 0 otherwise
k_{jtp} - quantity of product $p \in \mathcal{P}$ bought in spot market at node $n = (j, t) \in \mathcal{N}^{\mathcal{H}}$
l_{jtp} - quantity of product $p \in \mathcal{P}$ sold in spot market at node $n = (j, t) \in \mathcal{N}^{\mathcal{H}}$

A visualization of the variables just presented can be seen in Figure 4.1, where a vessel departs from port j in time period t and arrives in port k in time period t'.

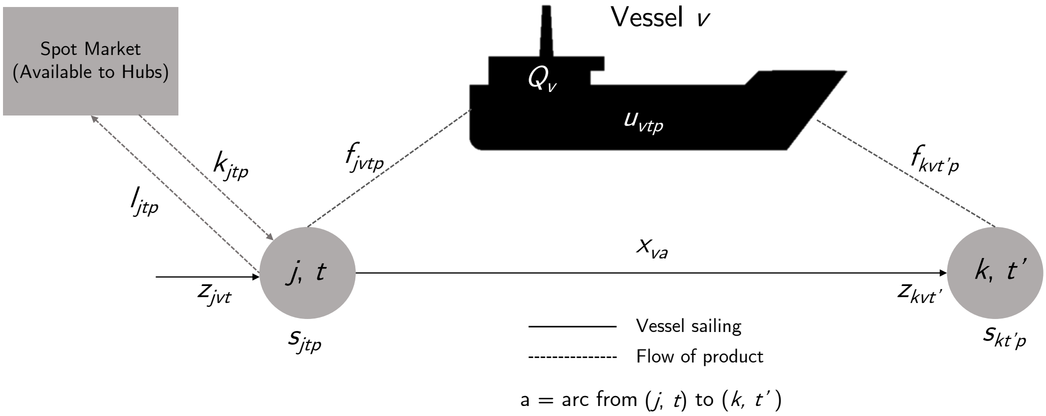

Figure 4.1: Visualization of variables.

For ease of exposition in the mathematical model, it is assumed that operation and travel for a vessel happens only in ports it is allowed to visit. Hence, all variables indexed by vessel v and port j are only defined for $j \in \mathcal{J}_v$.

4.1.3 Formulation

Objective Function:

$$\min \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_v} C_{va} x_{va} + \sum_{(j,t) \in \mathcal{N}^{\mathcal{H}}} \sum_{p \in \mathcal{P}} (C^B k_{jtp} - C^S l_{jtp}), \tag{4.1}$$

The objective function (4.1) minimizes the total cost and consists of two terms. The first term represents the total travelling cost, which is the sum of the cost of all arcs that are traversed by the vessels. The second term is the total cost from the spot market, which is the quantity bought minus the quantity sold multiplied with buying and selling price, respectively.

Flow Balance Constraints:

$$\sum_{a \in \delta^+_{vn}} x_{va} - \sum_{a \in \delta^-_{vn}} x_{va} = \begin{cases} 1 & \text{if } n = n_0 \\ -1 & \text{if } n = n_{(\mathcal{T})} \\ 0 & \text{if } n \in \mathcal{N} \end{cases}, \quad v \in \mathcal{V}, n \in \mathcal{N}_0 \tag{4.2}$$

$$z_{jvt} \leq \sum_{a \in \delta^-_{vn}} x_{va}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, \tag{4.3}$$

$$z_{jvt} \leq x_{va}, \quad v \in \mathcal{V}, (j,t) \in \mathcal{N}, a = \{(j,t),(j,t+1)\}, \tag{4.4}$$

$$z_{jv(t-1)} \leq z_{jvt} + \sum_{a=((j,t),(i,t')) \in \delta^+_{vn}, i \neq j} x_{va}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, t > 0 \tag{4.5}$$

Constraints (4.2) ensure that if a vessel enters a node it must also leave the same node. For the artificial start and end nodes, vessels will only leave or enter, respectively. Constraints (4.3) make sure that a vessel does not attempt to load or discharge at a node unless the vessel actually arrived at the node. In addition if the vessel is operating, it can not leave the port in the same time period due to constraints (4.4). Constraints (4.5) ensure that if a vessel operates it must either continue operating or leave in the next time period.

Inventory Balance Constraints:

$$s_{jtp} = s_{j(t-1)p} + D_{jtp} - \sum_{v \in \mathcal{V}_r} \Delta_j f_{jvtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{4.6}$$

$$s_{j0p} = S^0_{jp} + D_{j0p} - \sum_{v \in \mathcal{V}_r} \Delta_j f_{jv0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, p \in \mathcal{P}, \tag{4.7}$$

$$s_{jtp} = s_{j(t-1)p} + \sum_{v \in \mathcal{V}_0} f_{jvtp} - \sum_{v \in \mathcal{V}_r} f_{jvtp} + k_{jtp} - l_{jtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}^{\mathcal{H}} \cap \mathcal{J}_r, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{4.8}$$

$$s_{j0p} = S^0_{jp} + \sum_{v \in \mathcal{V}_0} f_{jv0p} - \sum_{v \in \mathcal{V}_r} f_{jv0p} + k_{j0p} - l_{j0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}^{\mathcal{H}} \cap \mathcal{J}_r, p \in \mathcal{P}, \tag{4.9}$$

$$s_{jtp} \geq (S^{max}_{jp} + S^{min}_{jp})/2, \quad n = (j,t) \in \mathcal{N}, p \in \mathcal{P}, t = |\mathcal{T}| - 1, \tag{4.10}$$

$$u_{vtp} = u_{v(t-1)p} + \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{P}}} f_{jvtp} - \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{H}}} f_{jvtp}, \quad v \in \mathcal{V}_0, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{4.11}$$

$$u_{v0p} = U^0_{vp} + \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{P}}} f_{jv0p} - \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{H}}} f_{jv0p}, \quad v \in \mathcal{V}_0, p \in \mathcal{P}, \tag{4.12}$$

$$u_{vtp} = u_{v(t-1)p} + \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{H}}} f_{jvtp} - \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{C}}} f_{jvtp}, \quad v \in \mathcal{V} \setminus \{\mathcal{V}_0\}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{4.13}$$

$$u_{v0p} = U^0_{vp} + \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{H}}} f_{jv0p} - \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{C}}} f_{jv0p}, \quad v \in \mathcal{V} \setminus \{\mathcal{V}_0\}, p \in \mathcal{P}, \tag{4.14}$$

Constraints (4.6) and (4.7) ensure inventory balance for all products in all ports that are not hubs when $t > 0$ and $t = 0$, respectively. Constraints (4.8) and (4.9) handle inventory balance in the hubs for $t > 0$ and $t = 0$. Inventory constraints for hubs are different, since vessels will both load and discharge at the hubs, which is not the case for production and consumption ports. Constraints (4.10) make sure that the inventory of ports for all products at the end of the planning period is greater than, or equal to, the average of the maximum and minimum storage capacity. Constraints (4.11) and (4.12) ensure inventory balance for vessels belonging to the production region for $t > 0$ and $t = 0$, respectively. These vessels load at production ports and discharge at hubs. Constraints (4.13) and (4.14) handle the inventory of vessels in the consumption regions for $t > 0$ and $t = 0$. These vessels load at hubs and discharge at consumption ports.

Quantity Constraints:

$$\sum_{v \in \mathcal{V}} z_{jvt} \leq B_j, \quad n = (j,t) \in \mathcal{N}, \tag{4.15}$$

$$F_j^{min} z_{jvt} \leq \sum_{p \in \mathcal{P}} f_{jvtp} \leq F_j^{max} z_{jvt}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, \tag{4.16}$$

$$S_{jp}^{min} \leq s_{jtp} \leq S_{jp}^{max}, \quad n = (j,t) \in \mathcal{N}, p \in \mathcal{P}, \tag{4.17}$$

$$0 \leq \sum_{p \in \mathcal{P}} u_{vtp} \leq Q_v, \quad v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.18}$$

Constraints (4.15) ensure that the total number of vessels loading or discharging at a node is not exceeding the berth capacity of the port belonging to that node. Constraints (4.16) make sure the quantity loaded or discharged of a product to or from a vessel is not above or below the upper and lower bound of loading capacities for the port. Additionally, loading and discharging can only happen at nodes that are actually visited by the vessel. Constraints (4.17) make sure that the quantity stored in a port is never below minimum or above maximum storage levels. Constraints (4.18) handle the capacity of the vessels.

Constraints on Variables:

$$0 \leq k_{jtp}, l_{jtp}, \quad (j,t) \in \mathcal{N}^{\mathcal{H}}, p \in \mathcal{P}, \tag{4.19}$$

$$0 \leq s_{jtp} \quad (j,t) \in \mathcal{N}, p \in \mathcal{P}, \tag{4.20}$$

$$0 \leq f_{jvtp} \quad (j,t) \in \mathcal{N}, v \in \mathcal{V}, p \in \mathcal{P}, \tag{4.21}$$

$$0 \leq u_{vtp} \quad v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.22}$$

$$x_{va} \in \{0,1\}, \quad v \in \mathcal{V}, a \in \mathcal{A}_v, \tag{4.23}$$

$$z_{jvt} \in \{0,1\}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, \tag{4.24}$$

Constraints (4.19) - (4.22) enforce non-negativity, while constraints (4.23) and (4.24) enforce binary requirements.

4.2 FCNF Formulation

To obtain better dual bounds than the Basic Arc-Flow formulation just presented, Agra et al. (2013) formulate a similar MIRP as a single-commodity fixed-charge network flow problem. In this formulation, the commodities supplied from the producer flow along the arcs corresponding to the vessel routes until they reach the consumption ports, where they are consumed externally. There are several known valid inequalities for FCNF problems, and applying these may further tighten the bounds.

In this formulation, each port-time is divided into three levels, which is presented in Figure 4.2. The first level deals with arrival and waiting of vessels. The middle level coordinates the vessel's operations and departures, and the bottom level is responsible for the inventory at the port. New sets, parameters and variables that are changed and added to the original Basic Arc-Flow formulation are given in Subsection 4.2.1. A mathematical formulation inspired by the FCNF formulation of Agra et al. (2013) is presented in Subsection 4.2.2, followed by tightening of variables and valid inequalities in Subsection 4.2.3 and Subsection 4.2.4, respectively.

4.2.1 Defintions

Sets

O_v - origin node for vessel $v \in \mathcal{V}$
D_v - destination node for vessel $v \in \mathcal{V}$

Parameters

Δ_v - an indicator parameter taking value $+1$ if $v \in \mathcal{V}_r$ where r is a consumption region, and -1 if $v \in \mathcal{V}_r$ where r is a production region.
T_{ijv} - travel time from port $i \in \mathcal{J}$ to port $j \in \mathcal{J}$ with vessel $v \in \mathcal{V}$. If $i = O_v$, T_{ijv} indicates the earliest available time for vessel $v \in \mathcal{V}$
$\bar{Q}_r$ - the largest vessel capacity of all vessels in region $r \in \mathcal{R}$

Variables

z^A_{jvt} - takes value 1 if vessel $v \in \mathcal{V}$ starts operating at port $j \in \mathcal{J}$ at time $t \in \mathcal{T}$, and 0 otherwise
z^B_{jvt} - takes value 1 if vessel $v \in \mathcal{V}$ continues to operate at port $j \in \mathcal{J}$ at time $t \in \mathcal{T}$ after starting to operate, and 0 otherwise
w_{jvt} - takes value 1 if vessel $v \in \mathcal{V}$ waits at port $j \in \mathcal{J}$ at time $t \in \mathcal{T}$, and 0 otherwise
u^X_{ijvtp} - load on board vessel $v \in \mathcal{V}$ of product $p \in \mathcal{P}$ when traveling from port $i \in \mathcal{J}$ to port $j \in \mathcal{J}$ leaving at time $t \in \mathcal{T}$
u^A_{jvtp} - load on board vessel $v \in \mathcal{V}$ of product $p \in \mathcal{P}$ when starting to operate in port $j \in \mathcal{J}$ in time period $t \in \mathcal{T}$
u^B_{jvtp} - load on board vessel $v \in \mathcal{V}$ of product $p \in \mathcal{P}$ before continuing to operate in port $j \in \mathcal{J}$ in time period $t \in \mathcal{T}$
u^W_{jvtp} - load on board vessel $v \in \mathcal{V}$ of product $p \in \mathcal{P}$ when waiting in port $j \in \mathcal{J}$ in time period $t \in \mathcal{T}$

The variable of operating, z_{jvt}, is now split into two variables, z^A_{jvt} and z^B_{jvt}. z^A_{jvt} indicates the start of operation in a port for a vessel in a given time period, while z^B_{jvt} indicates that an operation is continued after operating in the previous time period. Unlike the formulation of Section 4.1, there is now a separate waiting variable, w_{jvt}, instead of representing waiting as travelling to a node with the same port in a later time period. The flow variable keeping track of the inventory on vessels, u_{vtp}, is replaced by four flow variables that ensure correct flow of commodities on the arcs. The relation between these new variables is visualized in Figure 4.2 through an example. In this example, the vessel enters the port in the first time period and waits until the second period. After that it operates in two periods before it leaves the port.

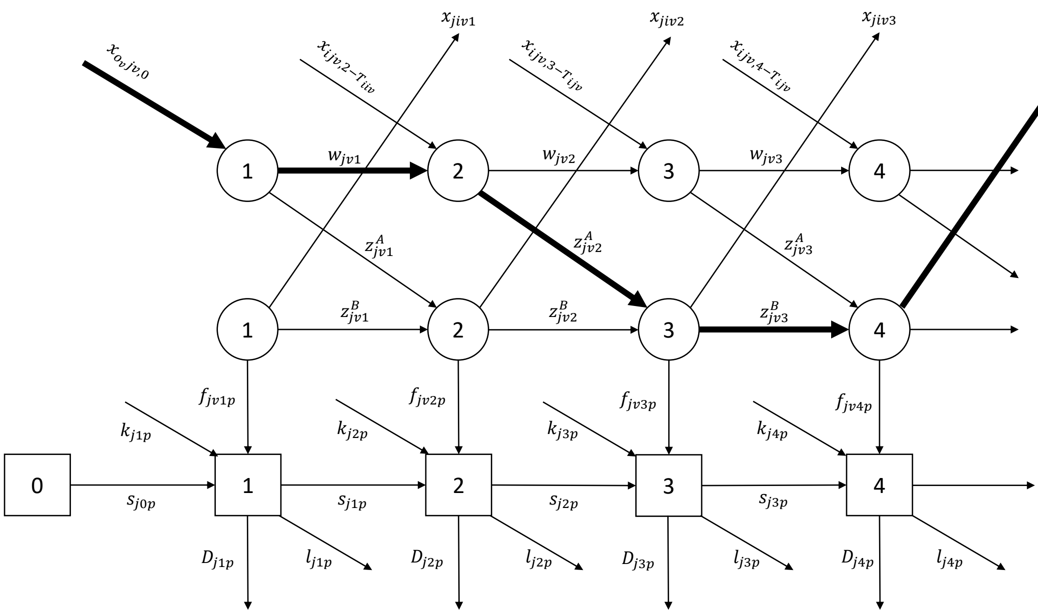

Figure 4.2: Visualization of the the three levels in each port in FCNF.

4.2.2 Mathematical Formulation

Objective Function:

$$\min \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{J}} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}} C_{ijv} x_{ijvt} + \sum_{j \in \mathcal{J}^{\mathcal{H}}} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} (C^B k_{jtp} - C^S l_{jtp}), \tag{4.25}$$

The objective function (4.25) is similar to the objective function (4.1), and minimize the total costs, consisting of travel cost and cost related to trade in the spot market.

Flow Balance Constraints:

$$\sum_{i \in \mathcal{J} \cup O_v} x_{ijv(t-T_{ijv})} + w_{jv(t-1)} = w_{jvt} + z^A_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \setminus \{0\}, \tag{4.26}$$

$$z^A_{jv(t-1)} + z^B_{jv(t-1)} = z^B_{jvt} + \sum_{i \in \mathcal{J} \cup D_v} x_{jivt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \setminus \{0\}, \tag{4.27}$$

$$\sum_{j \in \mathcal{J} \cup D_v} x_{O_v jv0} = 1, \quad v \in \mathcal{V}, \tag{4.28}$$

$$\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{J} \cup O_v} x_{iD_v vt} = 1, \quad v \in \mathcal{V}, \tag{4.29}$$

$$z^A_{jvt} + z^B_{jvt} = z_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.30}$$

The constraints (4.2) - (4.5) are now replaced by (4.26) - (4.30). Constraints (4.26) ensure that if a vessel arrives or waits in a port it must either wait or operate in the next time period. Constraints (4.27) make sure that if a vessel operates it must either continue to operate or leave the port in the next time period. All vessels must leave the source node, and enter the sink node, which is made sure by constraints (4.28) and (4.29). Connection between the binary variables related to operation is handled by constraints (4.30). These constraints describe the movement of the vessels in the extended network given in Figure 4.2.

Flow Balance Constraints on Vessels:

$$\sum_{i \in \mathcal{J} \cup O_v} u^X_{ijv(t-T_{ijv})p} + u^W_{jv(t-1)p} = u^W_{jvtp} + u^A_{jvtp}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{4.31}$$

$$u^A_{jv(t-1)p} + u^B_{jv(t-1)p} + \Delta_j f_{jv(t-1)p} = u^B_{jvtp} + \sum_{i \in \mathcal{J} \cup D_v} u^X_{jivtp}, \quad j \in \mathcal{J} \setminus \{\mathcal{J}^{\mathcal{H}}\}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.32}$$

$$u^A_{jv(t-1)p} + u^B_{jv(t-1)p} + \Delta_v f_{jv(t-1)p} = u^B_{jvtp} + \sum_{i \in \mathcal{J} \cup D_v} u^X_{jivtp}, \quad j \in \mathcal{J}^{\mathcal{H}}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.33}$$

$$u^X_{O_v jv0p} = U^0_{vp} x_{O_v jv0}, \quad v \in \mathcal{V}, j \in \mathcal{J} \cup D_v, p \in \mathcal{P}, \tag{4.34}$$

$$\sum_{p\in\mathcal{P}} u^X_{ijvtp} \leq Q_v x_{ijvt}, \quad v \in \mathcal{V}, i \in \mathcal{J} \cup O_v, j \in \mathcal{J} \cup D_v, t \in \mathcal{T}, \tag{4.35}$$

$$\sum_{p\in\mathcal{P}} u^A_{jvtp} \leq Q_v z^A_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.36}$$

$$\sum_{p\in\mathcal{P}} u^B_{jvtp} \leq Q_v z^B_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.37}$$

$$\sum_{p\in\mathcal{P}} u^W_{jvt} \leq Q_v w_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.38}$$

These constraints replace constraints (4.11)-(4.14) and (4.18). Constraints (4.31) ensure that the inventory on the vessel when it starts to operate or wait in a port is equal to the inventory the vessel arrived with, or the inventory that the vessel has been waiting with in the previous time period. The flow conservation on vessels in the production and consumption ports is maintained by constraints (4.32), and by constraints (4.33) for the hubs. The initial inventory on the vessels are set by constraints (4.34). The upper bounds on the flow variables are handled by constraints (4.35) - (4.38).

Inventory Balance Constraints:

$$s_{jtp} = s_{j(t-1)p} + D_{jtp} - \sum_{v\in\mathcal{V}_r} \Delta_j f_{jvtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{4.39}$$

$$s_{j0p} = S^0_{jp} + D_{j0p} - \sum_{v\in\mathcal{V}_r} \Delta_j f_{jv0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, p \in \mathcal{P}, \tag{4.40}$$

$$s_{jtp} = s_{j(t-1)p} + \sum_{v\in\mathcal{V}_0} f_{jvtp} - \sum_{v\in\mathcal{V}_r} f_{jvtp} + k_{jtp} - l_{jtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}^H \cap \mathcal{J}_r, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{4.41}$$

$$s_{j0p} = S^0_{jp} + \sum_{v\in\mathcal{V}_0} f_{jv0p} - \sum_{v\in\mathcal{V}_r} f_{jv0p} + k_{j0p} - l_{j0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}^H \cap \mathcal{J}_r, p \in \mathcal{P}, \tag{4.42}$$

$$s_{jtp} \geq (S^{max}_{jp} + S^{min}_{jp})/2, \quad j \in \mathcal{J}, p \in \mathcal{P}, t = |\mathcal{T}| - 1, \tag{4.43}$$

Constraints (4.39)-(4.43) are identical to (4.6)-(4.10).

Quantity Constraints:

$$\sum_{v \in \mathcal{V}} z_{jvt} \leq B_j, \quad j \in \mathcal{J}, t \in \mathcal{T}, \tag{4.44}$$

$$F^{min}_j z_{jvt} \leq \sum_{p \in \mathcal{P}} f_{jvtp} \leq F^{max}_j z_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.45}$$

$$S^{min}_{jp} \leq s_{jtp} \leq S^{max}_{jp}, \quad j \in \mathcal{J}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.46}$$

Constraints (4.44)-(4.46) are identical to (4.15)-(4.17).

Constraints on Variables:

$$0 \leq k_{jtp}, l_{jtp}, \quad j \in \mathcal{J}^{\mathcal{H}}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.47}$$

$$0 \leq s_{jtp} \quad j \in \mathcal{J}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.48}$$

$$0 \leq u^X_{ijvtp}, \quad i, j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.49}$$

$$0 \leq f_{jvtp}, u^A_{jvtp}, u^B_{jvtp}, u^W_{jvtp}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{4.50}$$

$$x_{ijvt} \in \{0, 1\}, \quad i, j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, t + T_{ij} \leq |T| - 1 \tag{4.51}$$

$$z_{jvt}, w_{jvt}, z^A_{jvt}, z^B_{jvt} \in \{0, 1\}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \tag{4.52}$$

Constraints (4.47)-(4.50) ensure that the variables take non-negative values. Binary requirements are enforces by constraints (4.51)-(4.52).

4.2.3 Tightening Flow Variables Bounds

Based on the requirements that a vessel must operate when visiting a port, and that a vessel must leave a port after operating, the upper and lower bounds of the flow variables can be tightened. Friske et al. (2021) demonstrate how the linear relaxation can be tightened and solutions can be obtained more efficiently with these bounds. Inspired by this work, we present tightening of flow variables for the problem in this thesis.

$$F_j^{min} z_{jvt}^B \leq \sum_{p \in \mathcal{P}} u_{jvtp}^B, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.53}$$

$$\sum_{p \in \mathcal{P}} u_{jvtp}^B \leq z_{jvt}^B (Q_v - F_j^{min}), \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.54}$$

$$z_{jvt} F_j^{min} \leq \sum_{p \in \mathcal{P}} (u_{jvtp}^A + u_{jvtp}^B), \quad j \in \mathcal{J}^{\mathcal{C}}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.55}$$

$$\sum_{p \in \mathcal{P}} u_{jvtp}^W \leq w_{jvt} (Q_v - F_j^{min}), \quad j \in \mathcal{J}^{\mathcal{P}}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.56}$$

$$w_{jvt} F_j^{min} \leq \sum_{p \in \mathcal{P}} u_{jvtp}^W, \quad j \in \mathcal{J}^{\mathcal{C}}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.57}$$

$$x_{ijvt} F_j^{min} \leq \sum_{p \in \mathcal{P}} u_{ijvtp}^X \leq x_{ijvt} (Q_v - F_i^{min}), \quad (i,j) \in \mathcal{J}^C, v \in \mathcal{V}, t \in \mathcal{T}, \tag{4.58}$$

Constraints (4.53) and (4.54) set lower and upper bounds on the inventory on a vessel when continuing to operate. Between two operations in a discharging port, the upper bound reflects the operation in the previous time period while the lower bound reflects the need for inventory in the next operation. For production ports, the same logic applies. Between two operations, the lower bound must reflect the minimum amount loaded in the previous operation and the upper bound ensures there is sufficient storage capacity on the vessel to load in the next operation. A vessel can only start operating in a consumption port if there is sufficient inventory, which is handled by (4.55). Upper and lower bounds on vessel inventory when waiting are set by (4.56) and (4.57) for production ports and consumption ports, respectively. Waiting is always followed by operation, and there must be sufficient inventory or storage capacity on the vessel to allow for this. Finally, constraints (4.58) handles lower and upper bounds on the vessel inventory when travelling between two consumption ports. The lower bound on inventory onboard must satisfy the minimum quantity that can be loaded in second port. The upper bound is affected by the operation in the first port.

4.2.4 Valid Inequalities

To obtain better bounds for the FCNF model, Agra et al. (2013) derives a set of different valid inequalities. These are based on knapsack sets, and are also presented in Friske and Buriol (2018) and Friske et al. (2021). Inspired by them, the following inequalities are adjusted to be applicable to the problem presented.

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} D_{jtp} + \sum_{p\in\mathcal{P}}(S_{jp}^{min} - S_{jp}^{max})}{\bar{Q}_0} \leq \sum_{i\in\mathcal{J}\cup D_v}\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} x_{jivt}, \quad j \in \mathcal{J}^P, \tag{4.59}$$

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} D_{jtp} + \sum_{p\in\mathcal{P}}(S_{jp}^{min} - S_{jp}^{max})}{F_j^{max}} \leq \sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} z_{jvt}, \quad j \in \mathcal{J}^P, \tag{4.60}$$

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} -D_{jtp} + \sum_{p\in\mathcal{P}}(S_{jp}^{min} - S_{jp}^{max})}{\bar{Q}_r} \leq \sum_{i\in\mathcal{J}}\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} x_{ijv(t-T_{jiv})}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \cap \mathcal{J}^C, \tag{4.61}$$

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} -D_{jtp} + \sum_{p\in\mathcal{P}}(S_{jp}^{min} - S_{jp}^{max})}{F_j^{max}} \leq \sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} z_{jvt}, \quad j \in \mathcal{J}^C, \tag{4.62}$$

Constraints (4.59) ensure the minimum number of vessel departures from the production port, which is the minimum amount of products that must be loaded from the port during the planning horizon divided by the maximum vessel capacity in the region. Similarly, constraints (4.61) enforce the minimum number of vessels arriving in consumption ports. The minimum numbers of operations required during the planning horizon are set by constraints (4.60) for production ports and constraints (4.62) for consumption ports. These are based on the minimum quantity that must be loaded or discharged in the planning horizon and the maximum rate of loading or discharging in the ports.

Chapter 5

Two-Stage Decomposition Matheuristic with Fix-and-Optimize

This chapter presents the matheuristic designed to solve the two-echelon multi-product maritime inventory routing problem described in Chapter 3. As the problem size increases, the problem becomes more difficult to solve, or even find a feasible solultion for, with exact solution methods. The main motivation for applying a matheuristic to the problem is to be able to efficiently find high-quality solutions for larger instances.

The problem studied in this master's thesis is unique compared to problems in the existing MIRP literature. As discussed in Chapter 2, the combination of two echelons, multiple products and MIRP is not present in the literature. In Section 2.6, several solution methods for 2E-IRP, MIRP and multi-product MIRP are presented and discussed, in addition to solution methods applied to combinations of these characteristics in problems. The matheuristic presented in this chapter is a new matheuristic inspired by the articles discussed, largely because of their good computational results.

The matheuristic presented in this chapter decomposes the full problem into smaller subproblems. These subproblems can be solved exactly, or with an additional heuristic for more complex subproblems. Section 5.1 gives an overview of the two-stage decomposition matheuristic applied to the full problem, in addition to the motivation and reasoning behind it. Section 5.2 presents the heuristic approaches applied as a solution method to complex subproblems, while Section 5.3 introduces the heuristic preprocessing applied to the full problem. The full matheuristic is hereby referred to as AD-RFFO, where AD refers to the aggregation of demand, and RFFO to the relax-and-fix and fix-and-optimize approach.

5.1 Two-Stage Decomposition Matheuristic with Feedback

As discussed in Chapter 3, each region has its own fleet of vessels. Therefore, taking advantage of the problem structure to create multiple independent MIRPs is the first step of the matheuristic. The fact that vessels in the production region supply the hubs is what prevents the consumption regions from being completely independent. A similar problem without the production region would in turn give multiple independent MIRPs. By decomposing the problem, and thus solving the production region first, each consumption region can be solved independently. As seen and described in Chapter 3 and Chapter 4, this is possible because the spot market is available to all hubs and ensures that all regional subproblems will be feasible. This approach has similarities to Papageorgiou et al. (2014b), who present a two-stage decomposition heuristic for a MIRP by aggregating the demand of multiple consumption ports into one super port, which resembles the regional hubs of the problem at hand. Applying this heuristic yielded good results on a set of public MIRP-instances. The heuristic outperformed commercial MIP-solvers, and is applicable to a two-echelon MIRP as the demand in the regions may be aggregated in the hubs.

Subsection 5.1.1 presents an overview of the matheuristic and how the problem is decomposed. Subsection 5.1.2 discusses and presents the changes necessary to the mathematical model to handle the decomposition of the problem, while Subsection 5.1.3 presents how the demand in the problems is aggregated as a part of the decomposition.

5.1.1 Overview

The aggregation-part of the matheuristic enables a decomposition of the problem, with each region constituting a subproblem. This is done by aggregating the demand of the consumption ports in each region into their respective hubs. The production region is therefore one subproblem, which is solved with the hubs as consumption ports with aggregated demand retrieved from the hub's respective consumption ports. The consumption regions are consequently solved with the hubs as production ports, and with the deliveries from the production region as production rates. The decomposition of the example problem illustrated in Chapter 3 is seen in Figure 5.1, which contains a total of four subproblems. Note that the *production stage* is the first stage of the decomposition, with only one production port and the regional hubs. The *consumption stage* is the next stage of the decomposition, where all regions with their regional hub and consumption ports constitute the consumption stage. Each region in this stage constitutes one subproblem each. Every subproblem in the consumption stage is solved independently. The problem to solve now consists of $|\mathcal{R}|$ smaller subproblems. However, an optimal solution for the full problem is not guaranteed with optimal solutions for each individual subproblem, and feedback must be implemented to create dependence between the subproblems in the production and consumption stage.

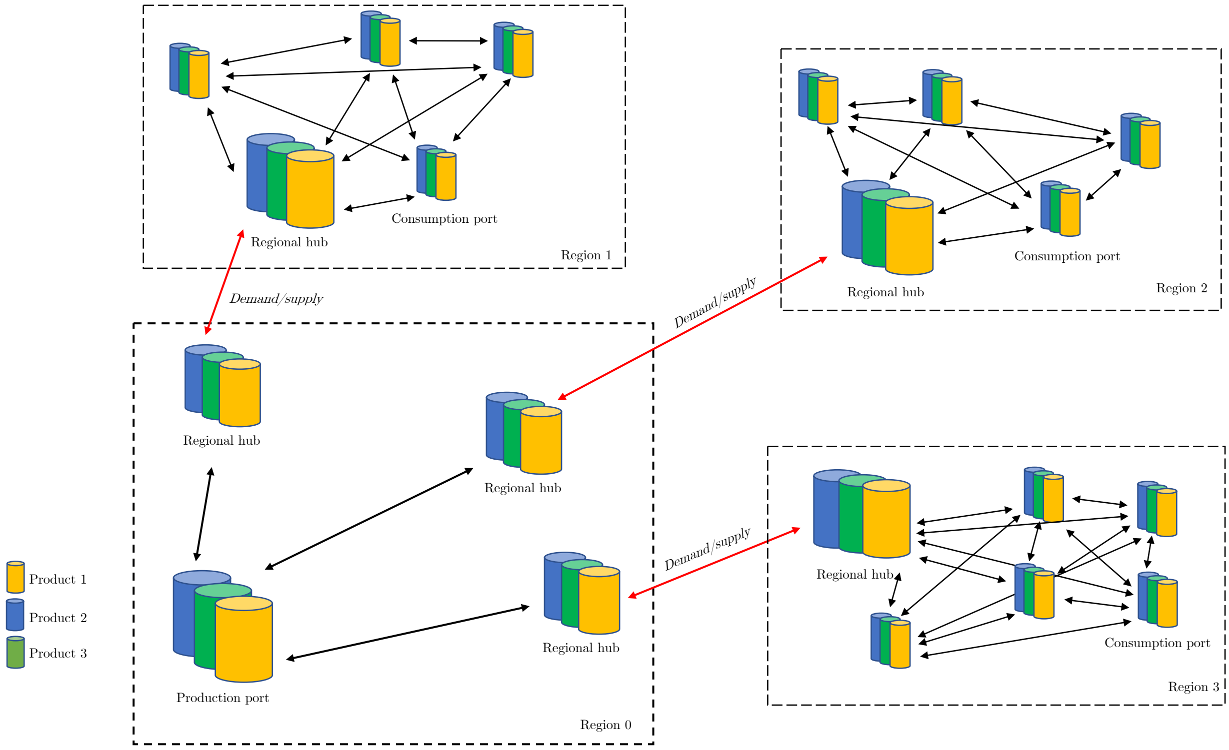

Figure 5.1: Example of problem-decomposition with production and consumption stages.

In addition to aggregating the demand in each region to the hubs, a parameter σ_j has been introduced to shift the demand in each consumption port by a specific set of time periods to take travel time in the consumption stage into account. This is done to ensure that products are available in a hub at the right time for shipment to its consumption ports. If the products arrive too late, the hub must buy in the spot market to fulfill the demand of the consumption ports. On the other hand, if products arrive too early, storage capacities may be exceeded and products must be sold in the spot market, which in turn might make the hubs buy products from the spot market at a later time.

The production and consumption stages are solved sequentially and in multiple iterations. The production region is solved first with the aggregated demand in the hubs, and then each of the consumption regions are independently solved with the deliveries from the production stage solution as production rates. One optimization of the production and consumption stage constitutes one iteration of the matheuristic, and the feedback is given as the parameter σ_j to shift the demand in the production stage.

The aggregated matheuristic with feedback is presented in Algorithm 1. The matheuristic iterates through all subproblems for as long as the total time limit is not reached, or as long as there is no change to σ_j. Note that *stageP* and *stageC* defines the production and consumption stages, respectively. The termination criteria can be seen in the while loop on line 5, and in lines 18-20.

Algorithm 1 Aggregated Matheuristic with Feedback

1: $\sigma \leftarrow$ INITIALSIGMA$(\mathcal{R}, \mathcal{J}, T_{ij})$
2: $\mathcal{M} \leftarrow$ set of all subproblems, $m_{StageP} \cup \mathcal{M}_{StageC}$
3: $m_{StageP} \leftarrow$ UPDATEMODEL(m_{StageP}, σ)
4: $bestObj \leftarrow \infty$
5: **while** $timeElapsed < totalTimeLimit$ **do**
6:     $sol_{StageP} \leftarrow$ OPTIMIZE(m_{StageP})
7:     initialize sol_{StageC}
8:     **for** m in $\mathcal{M}_{StageC}$ **do**
9:         $m \leftarrow$ UPDATEMODEL(m, sol_{StageP})
10:         $sol_{StageC}[m] \leftarrow$ OPTIMIZE(m)
11:     **end for**
12:     $obj \leftarrow$ OBJECTIVE$(sol_{StageP}, sol_{StageC})$
13:     **if** $obj < bestObj$ **then**
14:         $bestObj \leftarrow obj$
15:     **end if**
16:     $\sigma_{old} = \sigma$
17:     $\sigma \leftarrow$ UPDATESIGMA$(sol_{StageC}, \mathcal{J}, T_{ij})$
18:     **if** $\sigma_{old} = \sigma$ **then**
19:         **break**
20:     **end if**
21:     $m_{StageP} \leftarrow$ UPDATEMODEL(m_{StageP}, σ)
22: **end while**

The matheuristic initializes a starting σ_j on line 1 with the function INITIALSIGMA. The set of all subproblems in both the production stage and consumption stage is defined on line 2, with m_{StageP} being the only subproblem in the production stage, and M_{StageC} being the set of all subproblems in the consumption stage. The model for the production stage subproblem is updated in accordance with the new σ_j through UPDATEMODEL on line 3. The current best objective, *bestObj*, is set to infinity as a starting value. The iterations of the matheuristic begins on line 5. The matheuristic optimizes the production stage on line 6, and uses the solution sol_{StageP} to update the supply for the hubs in each of the consumption stage models on line 9. The optimization for each consumption stage subproblem is performed on line 10. Note that the list containing the solutions for all subproblems in the consumption stage sol_{StageC} is initialized on line 7. The function OPTIMIZE called on lines 6 and 10 is a solution method optimizing the model. OPTIMIZE has a time limit ($timeLimit_1$) for finding a first feasible solution, meaning that each subproblem has a given amount of time to find a feasible solution. Furthermore, OPTIMIZE implements a second time limit ($timeLimit_2$) when the first feasible solution is found. Note that the matheuristic saves the best objective value of all iterations, as there is a chance that an updated sigma will worsen the objective value. This is done on lines 12 - 15, with the current objective value calculated on line 12. The solutions found from the subproblems in the consumption stage is used to update σ_j through UPDATESIGMA on line 17. Finally, the model for the production stage is updated with the new σ_j on line 21 before the next iteration is started.

5.1.2 Mathematical Changes

To support the decomposition of the problem, changes must be made to the mathematical model presented in Chapter 4. The mathematical model has been split up into two types of models, one for each stage of the decomposition. The initial testing done in Section 6.3 showed that the FCNF-model is the most appropriate for the heuristic, so only the changes made to this model are presented in this subsection. The following changes can be applied in the same manner to the Basic Arc-Flow model.

For both types of models, hubs have been reassigned to consumption ports and productions ports, for the production and consumption stage respectively. Therefore, no ports are longer classified as hubs. This is done as the subproblems are not two-echelon, but single-echelon MIRPs. The mathematical models presented below all represent *one region* each, instead of the full problem as in Section 4.2. This means that $|\mathcal{R}| = 1$ for all models, and that all ports and vessels for other regions than the model's respective region are removed. Therefore, only one model of the production stage type is made, while $|R| - 1$ models of the consumption stage type are made. The following set is introduced:

$\mathcal{M}$ - set of all subproblems.

To represent the changes described above, subsets for each subproblem $m \in \mathcal{M}$ have been made for the sets of regions $\mathcal{R}$, vessels $\mathcal{V}$, production ports $\mathcal{J}^{\mathcal{P}}$, consumption ports $\mathcal{J}^{\mathcal{C}}$ and ports $\mathcal{J}$. The sets are defined as:

$\mathcal{R}_m$ - the respective region of subproblem m. $|\mathcal{R}_m| = 1$ for all $m \in \mathcal{M}$.
$\mathcal{V}_m$ - all vessels in $\mathcal{V}_r$, where $r = \mathcal{R}_m$.

$$\mathcal{J}^{\mathcal{P}}{}_m = \begin{cases} \mathcal{J}^{\mathcal{P}} & \text{if } m \text{ is a subproblem in the production stage.} \\ \mathcal{J}^{\mathcal{H}} \cap \mathcal{J}_r, r = \mathcal{R}_m & \text{if } m \text{ is a subproblem in the consumption stage} \end{cases}$$

$$\mathcal{J}^{\mathcal{C}}{}_m = \begin{cases} \mathcal{J}^{\mathcal{H}} & \text{if } m \text{ is a subproblem in the production stage.} \\ \mathcal{J}^{\mathcal{C}} \cap \mathcal{J}_r, r = \mathcal{R}_m & \text{if } m \text{ is a subproblem in the consumption stage.} \end{cases}$$

$\mathcal{J}_m$ - all ports in subproblem m, $\mathcal{J}_m = \mathcal{J}^{\mathcal{P}}{}_m \cup \mathcal{J}^{\mathcal{C}}{}_m$.

The following parameter has been added to both mathematical models to increase readability:

δ_j - takes value 1 if port j can buy or sell in spot market, 0 otherwise.

Production Stage

The parameter D_{jtp} is changed to represent the aggregated demand. The time-shift in demand for port j is defined as σ_j. D_{jtp} is updated according to the following equations:

$$D_{jtp} = \sum_{i \in \mathcal{J}^{\mathcal{C}} \cap \mathcal{J}_r} D_{i(t+\sigma_i)p}, \quad r \in \mathcal{R}, j \in \mathcal{J}^{\mathcal{H}} \cap \mathcal{J}_r, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P},$$

$$D_{j0p} = \sum_{i \in \mathcal{J}^{\mathcal{C}} \cap \mathcal{J}_r} \sum_{t=0}^{\sigma_i} D_{itp}, \quad r \in \mathcal{R}, j \in \mathcal{J}^{\mathcal{H}} \cap \mathcal{J}_r, p \in \mathcal{P},$$

Note that the sets $\mathcal{J}^{\mathcal{C}}$, $\mathcal{J}^{\mathcal{H}}$ and $\mathcal{J}_r$ are the original from Section 4.2. These must be used to aggregate the demand. Note that the second equation aggregates all demand where $t - \sigma_j < 0$ to $t = 0$, while the first equation handles all other t. The function UPDATEMODEL used in Algorithm 1 updates D_{jtp} as described, in accordance with σ_j.

The following is the mathematical model for subproblem m in the production stage:

$$\min \sum_{i \in \mathcal{J}_m} \sum_{j \in \mathcal{J}_m} \sum_{v \in \mathcal{V}_m} \sum_{t \in \mathcal{T}} C_{ijv} x_{ijvt} + \sum_{j \in \mathcal{J}^{\mathcal{C}}{}_m} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} (C^B k_{jtp} - C^S l_{jtp}), \tag{5.1}$$

s.t.

$$(4.23) - (4.29), (4.31) - (4.35), (4.40) - (4.58)$$

$$s_{jtp} = s_{j(t-1)p} + D_{jtp} - \sum_{v \in \mathcal{V}_m} \Delta_j f_{jvtp} + \delta_j(k_{jtp} - l_{jtp}), \quad j \in \mathcal{J}_m, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{5.2}$$

$$s_{j0p} = S^0_{jp} + D_{j0p} - \sum_{v \in \mathcal{V}_m} \Delta_j f_{jv0p} + \delta_j(k_{j0p} - l_{j0p}), \quad j \in \mathcal{J}_m, p \in \mathcal{P}, \tag{5.3}$$

The objective function (4.25) has been updated to (5.1) such that it sums over consumption ports when calculating the total cost of the spot market. Constraints (4.39) and (4.40) have been updated to include the spot market, as the spot market is preserved, but not the hubs. These are now constraints (5.2) and (5.3). The constraints control the inventory balance of all ports in the problem. The parameter δ_j is added to make sure that only the consumption ports are allowed to use the spot market.

Constraints (4.33) have been removed, as these are the flow conservation for hubs. The same applies to constraints (4.41) and (4.42), as these control the inventory balance for hubs. The only changes made to constraints (4.26) - (4.32), (4.34) - (4.38) and (4.43) - (4.62) are changing the sets $\mathcal{R}, \mathcal{V}, \mathcal{J}^{\mathcal{P}}, \mathcal{J}^{\mathcal{C}}$ and $\mathcal{J}$ to the respective sets indexed by subproblem m as presented above.

Consumption Stage

This mathematical model represents the consumption stage, with one mathematical model being created for each individual consumption region. The parameter D_{jtp} is here changed to represent the solution from the production stage, given by the f_{jvtp}-variables. Only the f_{jvtp}-variables from the production stage where j is a consumption port, originally classified as a hub, are kept. The amounts bought and sold in the spot market in the production stage are ignored, as the consumption stage will still make sure that all inventory balance constraints are satisfied. The final amount bought and sold will therefore be decided in each consumption stage to keep flexibility between the stages. Only the costs of the spot market from the consumption stage is included in the objective function for the full problem. With f^*_{jvtp} as the f-variables in the solution from the production stage, D_{jtp} is updated according to the following equation:

$$D_{jtp} = \sum_{v \in \mathcal{V}_m} f^*_{jvtp}, \quad m \in \mathcal{M}, j \in {\mathcal{J}^{\mathcal{P}}}_m, t \in \mathcal{T}, p \in \mathcal{P},$$

As there is only one production port in each region in the consumption stage, this will be done for all deliveries to this port in the solution from the production stage. The function

UPDATEMODEL used in Algorithm 1 updates D_{jtp} as described, in accordance with the solution from the production stage.

The following is the mathematical model for subproblem m in the consumption stage:

$$\min \sum_{i \in \mathcal{J}_m} \sum_{j \in \mathcal{J}_m} \sum_{v \in \mathcal{V}_m} \sum_{t \in \mathcal{T}} C_{ijv} x_{ijvt} + \sum_{j \in \mathcal{J}^{\mathcal{P}}{}_m} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} (C^B k_{jtp} - C^S l_{jtp}), \tag{5.4}$$

s.t.

$$(4.23) - (4.29), (4.31) - (4.35), (4.40) - (4.58),$$

$$(5.2) - (5.3),$$

The objective function (4.25) has been updated to (5.4) such that it sums over production ports when calculating the total cost of the spot market. Constraints (5.2) and (5.3) are the same as in the production stage, and constraints (4.33), (4.41) and (4.42) have been removed for the same reason as in the production stage. The only changes made to constraints (4.26) - (4.32), (4.34) - (4.38) and (4.43) - (4.62) are changing the sets $\mathcal{R}, \mathcal{V}, \mathcal{J}^{\mathcal{P}}, \mathcal{J}^{\mathcal{C}}$ and $\mathcal{J}$ to the respective sets indexed by subproblem m as presented above.

5.1.3 Time-shift in Demand

When aggregating the demand from a consumption port to a hub, the time at which the demand occurs must be shifted to take the travel time between the hub and the consumption port into account. As mentioned, this is done to ensure that the products arrive at the right time to the hub in the production stage.

A σ_j-parameter has been introduced to represent the shift in time for a port $j \in \mathcal{J}$. This parameter is updated between each iteration according to the consumption stage solutions, and is the feedback given from iteration i to iteration $i + 1$. As σ_j changes, the solution may change, giving a more suitable σ_j for the next iteration. An initial σ_j is calculated in the first iteration, and a new σ_j is calculated at the end of each iteration.

Initial σ_j

The initial σ_j is calculated as the total travelling time of all arcs in a region, but with each travel time between two ports just included once. Preliminary testing showed that the initial σ_j had little effect on the second iteration σ_j, and it is therefore calculated as a definite worst case instead of an arbitrary number. Using this method eliminates the need for a more complex algorithm for finding an initial σ_j. As ports in a real-life consumption region usually are close in proximity, the total travelling time will in most cases not be a large number. Preliminary testing of the initial σ_j showed that the parameter was always significantly less than the number of time periods, meaning that the time-shift was not severely exaggerated. The initial σ_j is the same for all ports $j \in J_r$ in a region r. With T_{ij} as the travel time between port i and port j, the initial σ_j is given as:

$$\sigma_j = \sum_{i \in \mathcal{J}_r} \sum_{k \in \mathcal{J}_r} \frac{T_{ik}}{2}, \quad j \in \mathcal{J}_r \cap \mathcal{J}^{\mathcal{C}}, r \in \mathcal{R} \setminus \{0\}$$

Each port in a region is assigned a σ_j equal to the time it takes to travel all arcs in the respective region *once*. The sum is therefore divided by two as all arcs are bidirectional. This is done for all regions except region 0, which is the production region as described in Chapter 4.

Updating σ_j

After each iteration of the matheuristic, σ_j must be updated in accordance with the consumption stage solutions. This may improve the solution of the next iteration. For every iteration except for the first one, σ_j represents the average time between each visit to port j. This is calculated as seen in Algorithm 2. Changes in σ_j also work as a termination criteria for the matheuristic. When σ_j has no change between two iterations, a local optima has been found and the matheuristic terminates.

In Algorithm 2, the consumption stage solutions are used to calculate the new sigma. Note that sol_{StageC} contains the solutions for each subproblem in the consumption stage. For each subproblem m, the algorithm finds a new σ_j for each consumption port in m. These loops can be seen on lines 2 and 3. A new σ_j for the consumption port, a counter for the number of visits to the port, and the total travel time between hub and the port are initiated on lines 4-6. To find the average time between deliveries, all deliveries to ports in the region must first be retrieved. Then, for all deliveries, the time difference between delivery and departure from the hub for the vessel which made the delivery is found. This can be seen on lines 7 - 11. More specifically, for each arrival into a consumption port j with vessel v, the previous departure from the hub with vessel v is found. By finding the time between the previous departure from hub and the arrival to port j for each incoming arc to port j, the average time between visits, and thus σ_j, is found. The new σ_j is calculated on line 15.

Algorithm 2 Updating σ_j

1: **function** UPDATESIGMA($sol_{StageC}, \mathcal{J}, T_{ij}$)
2:     **for** subproblem m in sol_{StageC} **do**
3:         **for** j in $m \cap \mathcal{J}_C$ **do**
4:             $\sigma_j \leftarrow 0$
5:             $nVisits \leftarrow 0$
6:             $totalTime \leftarrow 0$
7:             **for** each delivery of products to j in solution **do**
8:                 $t_{del} \leftarrow$ time of delivery of products to j
9:                 $t_{dep} \leftarrow$ previous departure from hub with the
10:                         same vessel as delivery to j
11:                 $travelTime \leftarrow t_{del} - t_{dep}$
12:                 $totalTime \leftarrow totalTime + travelTime$
13:                 $nVisits \leftarrow nVisits + 1$
14:             **end for**
15:             $\sigma_j \leftarrow totalTime/nVisits$
16:         **end for**
17:     **end for**
18:     return σ
19: **end function**

5.2 Cluster-Based RFFO Heuristic

As the total problem size grows, each of the subproblems in the consumption stage may become more complex, making them more difficult to solve exactly. Eventually, each subproblem requires a heuristic for being solved efficiently. As vessels in the production region are not allowed to travel between hubs, but only to and from the hub, the subproblem of the production region does not grow too complex for exact solution methods. This was seen in preliminary testing. With a high number of time periods, vessels, ports or products, each consumption region could constitute its own large and complex MIRP. To resolve this problem, a heuristic has been implemented to solve the subproblems in the consumption stage. As these regions are independent, the problems to solve are no longer two-echelon, and rather regular multi-product MIRPs. Solution methods applicable to these types of problems are discussed in Section 2.6, and *clustering* is a common decomposition approach in the literature to reduce the problem size, as seen in both Gaur and Fisher (2004) and Nambirajana et al. (2016). In addition, both R&F (relax-and-fix) and F&O (fix-and-optimize) heuristics as solution methods have yielded good results for IRPs and MIRPs, with Friske et al. (2021) obtaining good results on a MIRP using a combination of the two heuristics. The heuristic applied to complex subproblems, presented in this section, is a combination of these approaches. By clustering the ports in a consumption region, artificial regions are created. The clusters can be seen as artificial regions, as they share a similar structure with the regions in the full problem in terms of vessels not being allowed to travel directly between ports in regions, or clusters. These artificial regions can be solved efficiently with an R&F and F&O approach, hereby called *RFFO*, by letting each artificial region constitute a subproblem.

An overview of the clustering approach is outlined in Subsection 5.2.1, with the RFFO-heuristic applied to the clusters described in Subsection 5.2.2. The details of the R&F and F&O heuristics is presented in Subsection 5.2.3 and Subsection 5.2.4, respectively. Finally, the different methods of clustering are discussed in Subsection 5.2.5.

5.2.1 Overview of Clustering-Approach

The artificial regions, hereby called *clusters*, are created inside each consumption region. In practice, this means that ports are clustered, and arcs between clusters are removed. An example of the result of clustering applied to a region containing six consumption ports is presented in Figure 5.2. The approach taken for both R&F and F&O in this heuristic is to decompose the subproblems into clusters. One subproblem is therefore split up into $|C|$ smaller problems called *cluster-problems*. The smaller problems are called cluster-problems to separate them from the subproblems in the AD-RFFO. With a subproblem in the consumption stage as P, a smaller cluster-problem P_c for cluster c is defined as the full subproblem P where cluster c is the only cluster with binary requirements for variables. The rest of the clusters are either relaxed or fixed, depending on the phase of the heuristic. The heuristic applied to each subproblem can be split up into two phases: a construction phase and an improvement phase. The R&F-heuristic is applied in the construction phase, while the F&O-heuristic is applied in the improvement phase. As discussed, the subproblem of the production region does not grow too complex for exact solution methods. The cluster-based RFFO-heuristic is therefore solely applied to the consumption stage subproblems.

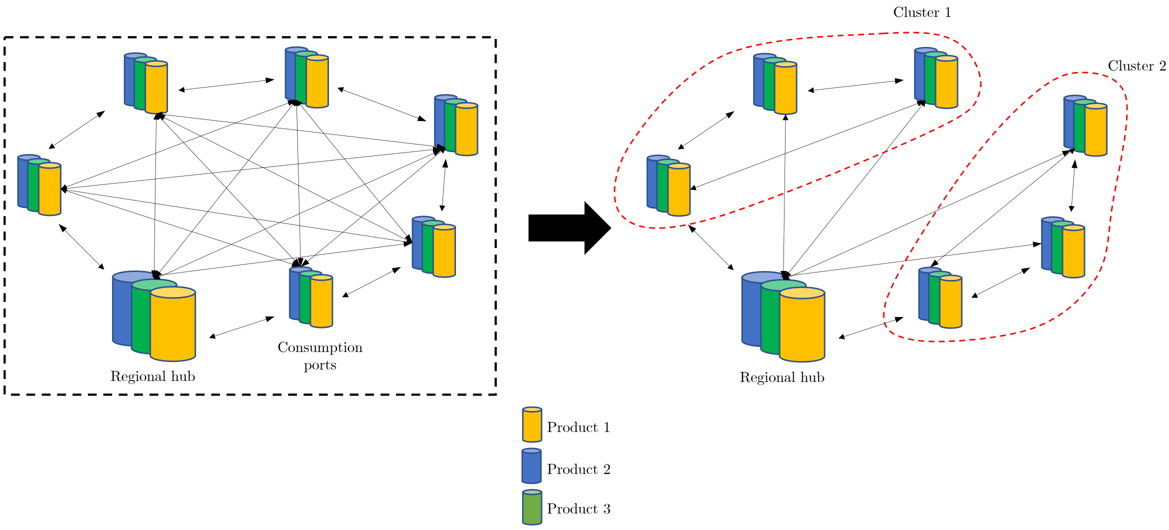

Figure 5.2: Region without clustering applied to the left, and with clustering applied to the right.

In addition to clustering the ports, each cluster is assigned a set of non-empty set of vessels. The vessels in this set are only allowed to use the arcs associated with the respective cluster. This constitutes arcs between ports in the cluster, and between ports in the cluster and the hub. The clusters are now more independent of each other, as they have their own fleet of vessels. The assignment of vessels to clusters are done as long as there are enough vessels so that each cluster is assigned the same number of vessels. More specifically, as long as $|C| \mod \mathcal{V}_{assigned} = 0$, where $|C|$ is the number of clusters and $\mathcal{V}_{assigned}$ is the number of vessels in the subproblem to be assigned to clusters. When this condition is no longer true, the remaining vessels will be *free vessels*: vessels that are allowed to operate in all clusters. This is done to make sure no cluster is assigned more vessels than others, and thus potentially making the clusters with fewer vessels infeasible. Vessels are assigned to clusters based on their capacity, which will be discussed in Subsection 5.2.5.

5.2.2 RFFO-Heuristic

Each region with clusters is solved by applying the RFFO-heuristic outlined in this section. An overview of the two phases is illustrated in Figure 5.3. In the first iteration, which is the construction phase applying an R&F-approach, binary requirements on variables in all clusters are relaxed. Thereafter, binary requirements are iteratively enforced to the respective clusters of the cluster-problems. For each cluster-problem, the full subproblem is solved, and the solution values are fixed for the respective cluster. When all clusters have their respective variables fixed to a solution, the improvement phase applying an F&O-heuristic begins. The variables of one cluster are unfixed, and subsequently solved together with the rest of the problem, which has fixed variables. The variables of the cluster are then fixed to the new solution. This is done iteratively for each cluster-problem. Note that a cluster will always be solved together with the rest of the relaxed or fixed clusters, for R&F or F&O respectively. Therefore, when a cluster-problem is solved, the full subproblem is also solved.

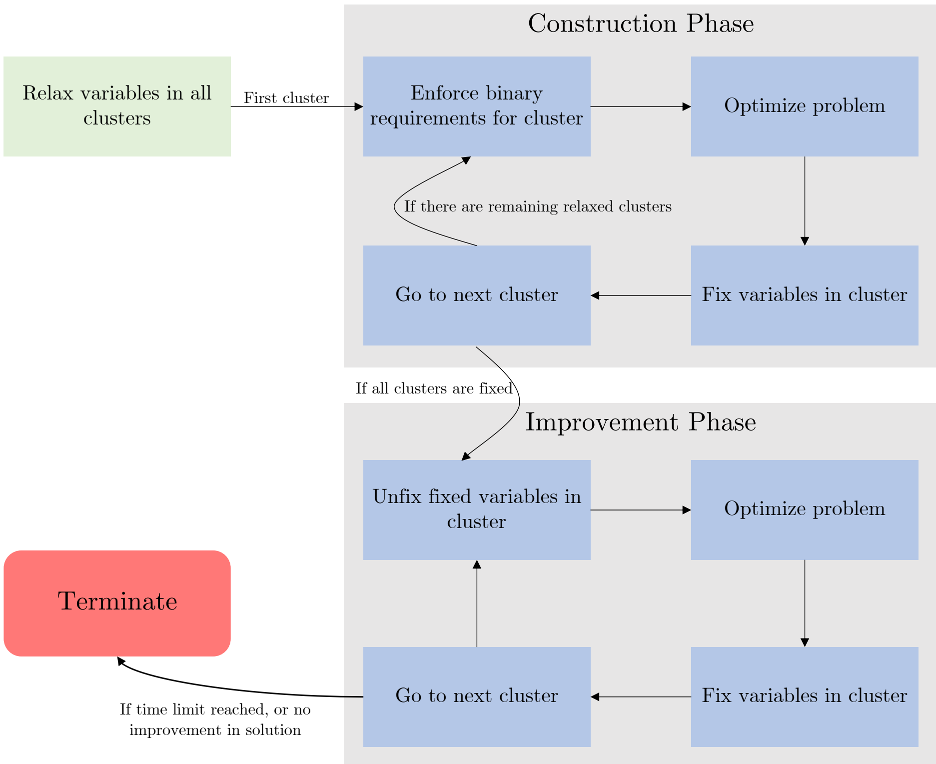

Figure 5.3: Illustration of the phases of the heuristic.

For instances applying the RFFO-heuristic, OPTIMIZE in Algorithm 1 calls the heuristic for each region in the consumption stage, as mentioned earlier. This is done instead of an exact solution method used for smaller instances. The construction and improvement phases are discussed in the next sections. See Appendix B.1 for a detailed pseudocode of the full RFFO-heuristic, where both phases are implemented as one compact RFFO-heuristic.

5.2.3 Construction Phase

The construction phase applies an R&F heuristic to generate a starting solution for the improvement phase. The main objective in this phase is to find a feasible solution. The details of this phase are outlined in Algorithm 3. Lines 2-6 relaxes all binary variables in subproblem m of the consumption stage. For the problem presented in this thesis, the binary variables relaxed in subproblem m are the x, z, z^A, z^B, and w-variables. These are the binary variables associated with using an arc, loading/discharging in a port, and waiting in a port, respectively. Lines 7-19 iterate through each cluster, and the variables belonging to the cluster are iterated through on lines 8-18. Lines 9-11 enforce binary requirements on the binary variables. Note that "originally binary" on line 9 indicates whether or not a variable var has binary requirements in the initial model. The full subproblem m is optimized using an exact solution method on line 13. This optimization runs until it finds a feasible solution, or it reaches a time limit $timeLimit_1$. If a feasible solution is found, the optimization restarts on line 14 with termination parameters $MIPGap$ and $timeLimit_2$, which are tolerances for the duality gap and computational time. On lines 15-18, variables are fixed to the solution found.

Algorithm 3 Construction Phase

```
function CONSTRUCTION(m)
    for var in m do
        if var is binary then
            var ← continuous
        end if
    end for
    for c in clusters do
        for var in c do
            if var is originally binary then
                var ← binary
            end if
        end for
        optimize m until feasible solution or timeLimit_1
        if feasible, optimize m until MIPGap or timeLimit_2
        for var in c do
            if var is binary then
                fix var to solution
            end if
        end for
    end for
    return solution of m
end function
```

The computational complexity of the problem is dominated by the binary variables to a much greater extent than the continuous variables. Therefore, extra flexibility is gained by not fixing any continuous variables. The initial solution found in the construction phase may be far from optimal for the full model, and an iterative process in the form of an improvement phase is appropriate.

5.2.4 Improvement Phase

To improve the initial solution found in the construction phase, an F&O heuristic is applied. The improvement phase iterates through all cluster-problems multiple times, in contrast to the construction phase with its single iteration. While the heuristic in the construction phase iterates through each cluster-problem once with the sole purpose of finding a feasible solution for the full subproblem, the F&O heuristic iterates through the cluster-problems until a local optimum is found or the time limit is reached. This can be seen in the while loop from line 3 to line 18 in Algorithm 4. Note that the initial current objective, *currentObjective*, is the objective from the solution in construction phase, as seen on line 1. A further distinction between the two phases lies in the name of the heuristics: while R&F relaxes all clusters and fixes one at a time, F&O unfixes and optimizes one cluster with the rest fixed. An initial previous objective, *prevObj*, is set to infinity on line 2 for starting the algorithm. For a given cluster, the heuristic unfixes the binary variables on lines 5-7, before optimizing the full model on lines 8-9. Note that the termination criterias are the same as described in Algorithm 3. On lines 11-13, the unfixed variables are fixed to the new solution. The variables to be fixed are the same in the improvement phase as in the construction phase, namely the binary x, z, z^A, z^B and w-variables. Finally, the previous and current objective values are saved on lines 16-17.

Algorithm 4 Improvement Phase

```
 1: function IMPROVEMENT(m, currentObj)
 2:     prevObj ← ∞
 3:     while currentObj < prevObj or elapsedTime < timeLimit do
 4:         for c in clusters do
 5:             for var in c do
 6:                 unfix var
 7:             end for
 8:             optimize m until feasible solution or timeLimit_1
 9:             if feasible, optimize m until MIPGap or timeLimit_2
10:             for var in c do
11:                 if var is binary then
12:                     fix var to solution
13:                 end if
14:             end for
15:         end for
16:         prevObj ← currentObj
17:         currentObj ← m.objective
18:     end while
19:     return solution of m
20: end function
```

5.2.5 Clustering Methods

Two different methods for clustering have been applied, and both are tested in Chapter 6. The two methods are based on physical distance between ports, and difference in demands for the ports.

Distance Clustering

When creating clusters based on physical distance between ports, two initial clusters are created, containing one consumption port each. One cluster contains the port furthest away from the hub, while the other contains the port closest to the hub. The remaining clusters are initialized by finding the ports evenly spaced in distance between the closest and most distant port. When all clusters have been initialized with one port each, the algorithm iterates through the clusters and finds the remaining port closest in distance to the initial port assigned to the cluster. This is done until all ports have been assigned to a cluster. Note that this method finds the ports closest to the initial port of a cluster, and not the port closest to the previous port assigned to the cluster. This method of clustering might not be an appropriate method if distances between ports are large, as it does not minimize the distance between all ports within a cluster. The last ports assigned to a cluster might be better assigned to another cluster. However, it is an effective method in the problem described in this thesis, as consumption regions contain ports that are in close to proximity each other. As travel distances are short, this method does most likely not produce unfavorable clusters.

When assigning vessels to clusters, the vessel with the highest capacity is assigned to the cluster with the initial port most distant from the hub. The vessel with the second highest capacity is assigned to the cluster with the second most distant cluster, and so on. For each iteration of assigning vessels to clusters, the most distant cluster gets the remaining vessel with the highest capacity. The free vessels not assigned to a cluster are therefore the vessels with the lowest capacities, as it is assumed that the free vessels have a smaller degree of utilization than vessels assigned to clusters. To ensure that the most distant clusters are the first to get the opportunity to utilize free vessels, these clusters are solved first.

Demand Clustering

Clustering based on the demand of the ports is done in the same way as with distances. The initial clusters are created based on the port with the highest and smallest total demand of the planning period, and the remaining clusters are initialized by the ports evenly spaced when sorted on total demand. The remaining ports are assigned to clusters in the same way as with distance, by iterating through each cluster and finding the port closest to the initial port in total demand. Instead of clustering based on creating an equal total demand between clusters, this is done to ensure the ports with the largest demand are served by large vessels. The vessels are assigned in the same way as with distance clustering: the vessel with the highest capacity is assigned to the cluster with the largest demand, and so on. By doing this, it is ensured that the ports with the largest demand both get the large vessels, and the opportunity to use what they need of potential free vessels before the smaller clusters are optimized. The cluster with the lowest amount of total demand are given the smallest vessel, and is also the last to use what is left of the potential free vessels.

5.3 Preprocessing

To reduce the total problem size, and thus more efficiently find good solutions, heuristic preprocessing inspired by Song and Furman (2013) and Friske et al. (2021) have been applied to the problem. Multiple preprocessing approaches are outlined in this section, and these are tested to find the best possible combination in Chapter 6.

The preprocessing seeks to remove variables that are unlikely to be necessary in an optimal, or near-optimal, solution. As binary variables have a much higher impact on the computational complexity than continuous variables, binary variables are the ones removed in the preprocessing. The x-variables make up the majority of total binary variables in the arc-flow model, while the z and w-variables are binary variables also affecting the computational time. The x-variables are indexed on vessel v, time period t, and ports i and j. The z and w-variables are indexed like the x-variable, but without the additional port i. The primary goal of the preprocessing is therefore to reduce the x-variables, and potentially some z and w-variables as a consequence of this.

Removing arcs based on time-periods

Two options of removing arcs based on time-periods are implemented and tested in this thesis. These include only being allowed to depart a port every *second time period*, or only being allowed to depart every *third time period*. As the planning period of MIRP's are long, it is assumed that reducing the possible times of departure is an effective way of reducing the number of x-variables without compromising the solution significantly. For the production region, where traveling times are long, vessels infrequently visit ports. Hence, the potential unnecessary waiting in ports is less significant. The negative effect of removing arcs in the consumption region can be greater, since the frequency of visits is much higher. Consequently, a compromise between reducing complexity and keeping the best solutions must be found with this preprocessing method.

Removing variables based on port-vessel combinations

Removing the possibility of a vessel in region r visiting a specific port in region r is another way of removing x-variables, and consequently removing z and w-variables. The ports removed for vessels in the production region are hubs, and consumption ports for the vessels in the consumption regions. These are all ports for discharging products in the full problem. Each vessel v has a set of such ports in $\mathcal{J}_v$ it is allowed to service, and by removing one unique discharging port j from each of the vessels' $\mathcal{J}_v$, one can also remove all x, z and w-variables connecting the vessel v to port j. Note that no port is removed from more than one vessel. For the production region, this can be effective as a vessel most likely never visit all hubs during the planning period. As travel between hubs are not allowed, removing a hub from a vessel's $\mathcal{J}_v$ will most likely not affect the solution value. For consumption regions, a vessel is not likely to visit all consumption ports between each visit to the hub, especially in instances with a high number of consumption ports. As there are multiple vessels in each consumption region and distances between consumption ports are short, removing a consumption port from a vessel's $\mathcal{J}_v$ is also a method that can prove effective and not remove optimal routes in a region.

This preprocessing is only applied where the number of vessels is larger than one, $|V| > 1$. This is done to ensure that all ports can be serviced. In addition, this is not applied if the cluster-based RFFO described in Section 5.2 is used. By applying both methods, a port in a cluster can potentially be removed from the only vessel servicing that specific cluster. Consequently, ports may end up not being allowed to be serviced by any vessels. This preprocessing method is done at a much larger scale in the clusters by removing whole groups of ports from vessels, namely the clusters.

Removing arcs based on distances

The final preprocessing approach removes arcs between *consumption* ports that are far away from each other. The distances between consumption ports are, as mentioned, short, but there are always some arcs that are longer than the rest. For each region, the longest travel time between two ports in the region is found, and all arcs with this travel time are removed. This is an aggressive preprocessing method potentially removing many x-variables, and it can prove highly effective in terms of computational complexity. It is however more prone to removing potential optimal solutions, especially where all ports in a region are close in proximity and a high number of arcs removed.

Chapter 6

Computational Study

This chapter presents the main findings from the computational study, which was conducted to develop the best version of the AD-RFFO matheuristic, presented in Chapter 5, and evaluate its performance compared to an exact solution method. Section 6.1 explains how test instances are generated, and the grouping of them is explained in Section 6.2. In Section 6.3, the performance of the mathematical models of Chapter 4 is discussed, and the one best suited for use in AD-RFFO, and for use in the final comparison to AD-RFFO, is identified. Following is Section 6.4, where preprocessing, clustering, and parameter tuning for the matheuristic are tested. Finally, Section 6.5 evaluates the performance of the AD-RFFO and compares it to the best exact solution method.

All models were run in Python 3.8.6 with the Gurobi Optimization version 9.1 solver. Gurobi was run using an academic license obtained via NTNU. The full computational study was conducted on NTNU's online computer cluster *Solstorm* [1]. There are multiple different node racks in Solstorm, and all tests in this study was run on nodes 4-16 to 4-20 in node rack 4. The nodes run on a Lenovo NextScale nx360 M5, with a 2 x 12-core Intel @ 2.3GHz processor and a memory of 64 GB RAM.

6.1 Test Instance Generation

As discussed in Section 2.7, the MIRP literature does not contain any studies of two-echelon problems. Consequently, there are no publicly available benchmark instances for testing and comparing the performance of the matheuristic and exact models presented in this thesis. Test instances were generated with the aim of representing real world maritime supply chains with two echelons. Larger vessels transport products across continents to regional hubs. Smaller vessels will collect the products at the hubs and distribute to destinations located in close proximity to the regional hub.

Each test instance is generated by first drawing both distances between ports and vessel capacities randomly from predefined intervals, which are based on realistic data. Re-

[1] https://solstorm.iot.ntnu.no/wordpress/

maining parameters are, at least partly, functions of vessel capacities and/or distances to ensure realistic proportions within each instance. The following sections explains in detail how parameters are generated.

6.1.1 Vessels

Data on vessels is extracted from Yieldstreet (2019) and Tankers (2014), and different vessel types are retrieved to reflect real life. For the production region, the vessel capacity is drawn in the interval [120.000 dwt, 200.000 dwt] based on the vessel type *Suezmax*. Vessel capacity in the consumption region is based on the vessel type *LR1* and takes values in the interval [55.000 dwt, 80.000 dwt]. The vessel speed is set to 15 knots for all vessels. Fuel consumption is 40 and 60 metric tons per day for vessels in consumption and production region, respectively.

To simulate a realistic initial state, some non-empty vessels in the production region are available at a given number of time periods after the start of the planning horizon. The aim of this adaptation is to emulate the movement of vessels that departed from ports before the start of the planning period. More specifically, $\lfloor \frac{N^{VP}}{2} \rfloor$ vessels will be available between 5 and 15 time periods after the start of the planning horizon, where N^{VP} is the number of vessels in the production region. Each vessel is loaded with 50% of total capacity with equal quantities of each product. Due to the short sailing distances in the consumption regions, all vessels in these regions are available at the start of the planning horizon with no initial storage.

6.1.2 Distances

Distances between the production port and hubs represent deep sea shipping. The maximum distance of 8000 nautical miles is inspired by distance between Yanbu in the Middle East and Houston in the US. The minimum distance is set to 4000 nautical miles and represents a trip from the Middle East to Europe. Hence, the distance between the production plant and each hub is between 4000 and 8000 nautical miles, resulting in sailing times between 11 and 22 days. An online sea distance calculator from ShipTraffic (2022) was used to find distances. Distances within regions, meaning between hubs and consumption ports and between consumption ports, represent short sea shipping for the final distribution. The interval is 250 to 1000 nautical miles, translating to travel times between 1 and 3 days.

6.1.3 Production and Demand

To avoid substantial discrepancy between transportation capacity and quantities demanded, the demand in a port is a function of the fleet capacity in the region it belongs to. It is assumed that a port can be visited every 20th day on average. The daily demand D_{jtp} for a product in a port should on average use the total vessel capacity of the region every 20th day, divided by the number of ports in the region and number of products.

In each region r, D_r^{MP} is the midpoint of the interval from which the daily demand is drawn. The following parameters were used to calculate it: the smallest capacity among all vessels in the region Q_r^{min}, the number of vessels in region N_r^V, the number of products N^P, and the number of ports per region N^J. The midpoint is calculated as follows:

$D_r^{MP} = \frac{Q_r^{min} N_r^V}{20 N^P N^J}$

To allow some variability between ports, the lower and upper limit for the demand per product in a port in region r has a 25% deviation from the average value calculated as described above. The interval for D_{jtp} where port j belongs to region r is $[0.75 D_r^{MP}, 1.25 D_r^{MP}]$. Daily production of a product in the production port is the sum of the daily demand of that product from all consumption ports.

6.1.4 Port capacities

Port storage capacities are functions of average daily demand or production, and take travel times into considerations. The initial storage in all ports is the average of minimum and maximum storage capacity. In a consumption port j, the minimum storage for a product p, S_{jp}^{min}, is 5 times the average daily demand of p, while the maximum storage S_{jp}^{max} is 30 times the average daily demand. Hence, the consumption ports must be replenished at least every 25th day. Due to the long sailing distances in the production region, the storage capacity in the production port must allow for more infrequent visits from the vessels of the region. Hence, S_{jp}^{min} and S_{jp}^{max} are 10 and 70 times daily production, respectively. Hubs are connected to the spot market and are less reliant on its own storage. For hubs, S_{jp}^{min} and S_{jp}^{max} are 0 and 15 times the total of the average daily demand for all ports in the region, respectively.

The loading and discharging capacity for ports in a region is related to the maximum capacity of the largest vessel in the same region, Q_r^{max}. A maximum loading and discharging capacityn in a port j, F_j^{max}, in region r is drawn randomly from $[0.5 Q_r^{max}, Q_r^{max}]$, while the minimum capacity, F_j^{min}, is drawn from $[0.01 Q_r^{max}, 0.05 Q_r^{max}]$.

The berth capacity for consumption ports is a random integer between 1 and the number of vessels in a consumption region N^{VC}. Similarly, the berth capacity of the production port is between 1 and the number of vessels in the production region N^{VP}. Hubs are served by vessels from both the production region and one consumption region. To reflect a higher frequency of visit in these ports, berth capacity is set between N^{VC} and $N^{VC} + N^{VC}$ for hubs.

6.1.5 Spot Market Price

The spot market is intended to be used mainly in situations where there is temporarily insufficient amount of goods flowing from the production port to a hub. To discourage excessive use of the spot market, the price of buying a given quantity must be higher than the theoretical maximum cost of moving the same quantity from production port to a hub. The cost of buying in the spot market C^B is calculated as $C^B = \frac{T^{max} F^{VP} C^F}{Cap^{min}}$, where T^{max} is the maximum sailing time from production port to hub, F^{VP} and C^F is the fuel consumption and cost, and Cap^{min} is the smallest capacity of a vessel in the production region. The fuel cost is set to \$600/mt. It is here assumed that vessels are fully loaded. The selling price must be lower than the buying price to avoid unbounded problems. It is set to half the buying price.

6.2 Grouping of Test Instances

There are multiple parameters subject for variation in each test instance, and a suitable grouping is necessary in order to perform a thorough and systematic computational study. An instance has been created as a base for each of the groupings. This base instance has $|\mathcal{T}| = 40$ time periods, $|\mathcal{R}| = 3$ regions, $|\mathcal{J}| = 3$ consumption ports in each region, $|\mathcal{VP}| = 3$ vessels in the production region, $|\mathcal{VC}| = 3$ vessels in each of the consumption regions and $|\mathcal{P}| = 2$ products, as presented in Table 6.1. Based on this, new instances are created by reducing or increasing one dimension at a time.

Table 6.1: Base instance B1.

Group	**Instance number**	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{VP}\rvert$	$\lvert\mathcal{VC}\rvert$	$\lvert\mathcal{P}\rvert$
Base	B1	40	3	3	3	3	2

The instances are grouped based on which parameter is changed in each test instance. The changes are made in number of time periods, regions, ports, vessels and products. Changes in different parameters make up the different groups. As an example, the group named *Time* will only have instances that differ in number of time periods, while keeping all other parameters constant. In all but one of the instance groups, the demand is kept constant, and it is not changed during the planning period. In order to test whether a fluctuating demand influences performance of the solution methods, a group with variable demand is also created. Two groups of complex instances are also created to test the performance of the solution methods when the instances become more complex. For the complex instances, changes are made in multiple dimensions at a time. The changes are primarily made in the number of ports, the number of time periods and regions to increase the total instance size.

The range in parameters for the test instances of each group is presented below in Table 6.2, where the smallest and the most complex instance in each group is presented. A full table with all test instances is given in Appendix C.1. The purpose of grouping the instances in this manner is to systematically test the effect of changes in one parameter at a time. This allows us to better analyze how the different parameters affect the complexity of the problem. A total of 75 instances are subject to testing, and the main findings are presented in Section 6.5.

The general rule is that the number of vessels in the production region, $|\mathcal{V}P|$ is the same as in consumption regions, $|\mathcal{V}C|$. In some cases, when the complexity increases, the number of vessels needed in the production region also increases in order for the instance to be feasible. The main reason for this being that when the number of consumption ports grow, the production rate also grows. The storage capacity of the production port is set according to this rate, but the vessel capacities are constant. To accommodate the

Table 6.2: Range of all groups of test instances.

Group	Instance no.	$\|\mathcal{T}\|$	$\|\mathcal{R}\|$	$\|\mathcal{J}\|$	$\|\mathcal{V}P\|$	$\|\mathcal{V}C\|$	$\|\mathcal{P}\|$
Time	T20	20	3	3	3	3	2
	T200	200	3	3	5	3	2
Var. demand	VD1	20	3	3	3	3	2
	VD10	200	3	3	5	3	2
Regions	R2	40	2	3	3	3	2
	R30	40	30	3	15	3	2
Ports	J1	40	3	1	3	3	2
	J8	40	3	8	3	3	2
Vessels	V1	40	3	3	1	1	2
	V8	40	3	3	8	8	2
Products	P1	40	3	3	3	3	1
	P8	40	3	3	3	3	8
Complex 1	C1.1	40	4	6	8	3	3
	C1.10	80	4	8	8	4	3
	C2.1	50	5	6	3	3	2
Complex 2	C2.9	100	5	9	10	5	2
	C2.10	40	5	12	6	6	2

increased production and consumption rates, the number of vessels in production region is also subject for change in larger instances.

To conduct the initial testing in Section 6.3 and Section 6.4, a group of 20 initial instances, containing a subset of three or four instances from each group in Table 6.2, was created. The full set of initial test instances can be seen in Appendix C.2. Additionally, the instance group *Complex 2* was used as the 10 initial complex instances in Section 6.4 for testing the performance on more computationally heavy instances.

6.3 Initial Testing of Exact Models

There were two primary goals with the initial testing of the exact models in this section. The first goal was finding the mathematical model for use in the matheuristic. To solve the subproblems of the decomposed problem, the best of the two mathematical models presented in this thesis is used as an exact solution method unless clustering is applied. The second goal was identifying the best mathematical model for finding good primal and dual solutions when solving larger problem instances exactly, which would be used in the comparison with the AD-RFFO in Section 6.5. The goal of AD-RFFO, presented in Chapter 5, is to provide high quality solutions for problem instances which can not be solved by a commercial solver. As the problem has a long planning horizon, computational time of multiple hours can be accepted. However, preliminary testing has shown that the objective value is to a small degree improved with computational times beyond two hours. Consequently, the mathematical model for solving larger instances exactly and the matheuristic is adapted to find the best objective values within a time frame of two hours.

As described in Section 5.1, the matheuristic splits the entire problem into many subproblems, which are solved sequentially. In most cases, each subproblem is solved multiple times. Consequently, the main requirement for the mathematical model solving these subproblems is reliably finding feasible, high quality solutions in a short amount of time. To reflect this requirement, 600 seconds were chosen as the computational time limit when testing was conducted to find the mathematical model to use in the matheuristic. When testing was conducted to find the best model for solving larger instances exactly, the computational time was increased to one hour. Although two hours is the time limit in the final comparison of Section 6.5, one hour is assumed to give an accurate picture of the relative performance and was chosen due to the high number of initial tests to be run. In Subsection 6.3.1, a comparison of the two mathematical models from Chapter 4 is presented, followed by Subsection 6.3.2 where combinations of valid inequalities and variable bound tightening for the FCNF formulation are tested.

6.3.1 FCNF vs. Basic Arc-Flow Model

Table 6.3 presents the results most relevant for evaluating which of the models, Basic Arc-Flow or FCNF, is best suited for use in the matheuristic, where the objective is reliably finding feasible, high quality solutions within a short amount of time. The number of instances where the models obtained a feasible solution within 600 seconds is reported (No. of sol.), as well as the number of times where the configuration had the best objective value among both models after 600 seconds (No. of best). In cases where the two models found solutions with less than 0,01% deviation in objective value, both were considered best. Finally, the average time to first feasible solution is reported (Avg. time feas.). Averages are solely based on instances where both models found feasible solutions within 3600 seconds. Based on the results from table Table 6.3, it is evident that the FCNF is substantially better than the Basic Arc-Flow model at finding feasible solutions of high quality in a relatively short amount of time. Hence, it is the choice of model for use in the matheuristic. This is illustrated by the chosen configuration highlighted in green in the table.

Table 6.3: Results from test conducted on 20 initial instances to find the exact model for use in the heuristic. No. of sol. and no. of best reported after 600s.

	FCNF	**Basic Arc-Flow Model**
No. of sol.	14	6
No. of best	14	4
Avg. time feas. [s]	154	825

The second purpose of the initial testing was deciding which of the two exact models had the best performance when the computational times increased, where the goal was reliably finding good primal and dual solutions. The first two rows of Table 6.4 reports the average objective value (Avg. obj. value) and dual gap as reported from Gurobi (Avg. gap) after 3600 seconds for 12 out of 20 instances, where both models found a feasible solution. Similar to Table 6.3, the number of best solutions is reported, now with a computational time of 3600 seconds. Finally, the number of instances where a model found the highest dual bound (No. of best bound) is reported. In accordance with the results of Friske et al. (2021), the dual bounds of the FCNF model is significantly higher than those of the Basic Arc-Flow model. Additionally, the FCNF model provides best solutions on most instances and has a better average objective value than the Basic Arc-Flow model. Consequently, the FCNF model was chosen as the exact solution method to be compared to AD-RFFO, which will be explored further in Section 6.5.

Table 6.4: Results from both exact models tested on the 20 initial instances, with focus on longer computational times. Results reported after 3600s.

	FCNF	**Basic Arc-Flow Model**
Avg. obj. value	3 578 557	3 649 784
Avg. gap	7,68%	89,17%
No. of best	16	5
No. of best bound	20	0

6.3.2 Testing of FCNF Configurations

This subsection presents results from the testing of different configurations of the variable bound tightenings and valid inequalities for the FCNF model, presented in Subsection 4.2.3 and Subsection 4.2.4, respectively. In order to find the best configuration, valid inequalities and variable bound tightenings were grouped according to their properties, and presented in Table 6.5. The "F" in the naming represents the FCNF model, while "V" and "T" are valid inequalities and variable bound tightenings, respectively. "FC" is a combination of the best set of valid inequalities and the best set of tightenings. The complete set of configurations, as well as the constraints they represent, can be seen in Table 6.5.

Table 6.5: Groups of valid inequalities and variable bound tightenings.

Mathematical model	Configuration	Name
FCNF	-	FCNF
FCNF + VI	(4.55), (4.57)	FV1
	(4.56), (4.58)	FV2
	(4.55) - (4.58)	FV3
FCNF + Tightening	(4.49) - (4.51)	FT1
	(4.52) - (4.53)	FT2
	(4.54)	FT3
	(4.49) - (4.53)	FT4
	(4.49) - (4.51), (4.54)	FT5
	(4.52) - (4.54)	FT6
	(4.49) - (4.54)	FT7
FNCF Combo	(4.49) - (4.51), (4.55), (4.57)	FC

Each group and combination of groups were tested to find the best final configuration. First, the FCNF model without any valid inequalities or tightenings was run to obtain benchmark results, as can be seen in Table 6.6. The best combination of valid inequalities was then determined by testing and comparing FV1, FV2, and FV3. Tightenings of variable bounds were then tested, namely the combinations FT1 - FT7. Finally, the best among FV1 - FV3 was combined with the best among FT1 - FT7, forming FC.

Similar to Table 6.3 from the previous subsection, Table 6.6 presents the number of feasible and best solutions after 600 seconds, as well as the average time to first feasible solution. With the aim of reliably finding high quality solutions within 600 seconds, FV1 stands out as the best alternative for use in the matheuristic with the best score on all metrics presented in the table. FV1 are valid inequalities setting lower bounds on the minimum number of vessels arriving to or departing from a port during the planning horizon.

Table 6.6: Results from the FCNF configurations tested on the 20 initial instances, with focus on matheuristic configuration. No. of sol. and no. of best reported after 600s.

	No. of Sol.	**No of Best**	**Avg. Time Feas. [s]**
FCNF	14	3	233
FV1	16	5	150
FV2	15	3	247
FV3	13	2	388
FT1	15	3	297
FT2	13	2	399
FT3	13	3	406
FT4	13	3	322
FT5	12	3	439
FT6	12	3	399
FT7	14	0	226
FC	14	3	156

Table 6.7 presents the results from the tests conducted to find the best combination of valid inequalities and variable bound tightenings for longer computational times in the comparison with the AD-RFFO. It reports the average objective value and gap on the 14 out of 20 instances to which all configurations found a feasible solution within the time limit of 3600 seconds. The numbers of best objective values and bounds are calculated similarly as the data of Table 6.4. Although the main emphasis is put on finding the best primal solutions, higher dual bounds can also help evaluate the performance of the matheuristic. With a good trade-off between primal and dual solutions, FV1 is the preferred configuration for longer computational times.

Table 6.7: Results from the FCNF configurations tested on the 20 initial instances, with focus on longer computational times. Results reported after 3600s.

	Avg. Obj. Value	Avg. Gap	No. of Best	No. of Best Bound
FCNF	4 605 417	7,21%	8	5
FV1	4 596 418	6,79%	7	8
FV2	4 604 318	6,64%	5	10
FV3	4 601 892	7,05%	5	5
FT1	4 611 353	7,41%	6	4
FT2	4 601 907	6,99%	6	5
FT3	4 601 002	6,18%	8	5
FT4	4 622 435	7,04%	5	3
FT5	4 600 173	6,93%	7	2
FT6	4 615 737	7,44%	4	3
FT7	4 622 723	7,57%	4	6
FC	4 613 726	7,39%	7	3

6.4 Initial Testing and Tuning of AD-RFFO

From the results of Section 6.3, it became clear that the FCNF model with the configuration previously defined as FV1 was to be used in the AD-RFFO. This section presents the testing conducted to develop a matheuristic with the ability of solving large problem instances within two hours. More specifically, preprocessing and clustering techniques were tested, as well as different parameter settings. Subsection 6.4.1 reports testing of the preprocessing techniques from Section 5.3. With the best preprocessing technique applied, the two clustering methods of Section 5.2 are tested in Subsection 6.4.2. With preprocessing and clustering applied, the gap and time limits are subject to tuning in Subsection 6.4.3.

The 10 initial complex instances were run in all tests to evaluate the performance on larger instances. To be able to solve these instances, clustering was applied in the form of distance clustering. Preliminary testing showed that distance clustering worked well, and this was chosen as the standard before the testing of different clustering methods. Additionally, the same 20 initial instances as in Section 6.3 were used, in order to have benchmark results from the exact FCNF model. As will be explained shortly, clustering is not active unless there are at least 6 ports in each region. Hence, the 20 initial instances, which all have less than 6 ports, were not run when comparing clustering methods.

The maximum computational time was 7200 seconds for the 10 initial complex instances, similar to the final testing in Section 6.5. The time limit of 3600 seconds, as previously used for the 20 initial instances, was unchanged. Note that the total elapsed time for AD-RFFO is checked prior to the start of each iteration. In a small number of cases this has caused the total computational time to slightly exceed the time limit, as a final iteration can be initiated just before the time limit is reached.

Prior to the parameter tuning in Subsection 6.4.3, initial values based on preliminary testing were set, as reported in Table 6.8. $MIPgap$ refers to the dual gap tolerance before termination in each subproblem or cluster-problem. $timeLimit_1$ is the total time limit for each subproblem and cluster-problem. It is not subject to tuning, as it is assumed that if no feasible solution is found in *one* of the subproblems or cluster-problems within the given time limit, the full problem is too complex to be solved within the total time limit as multiple iterations are necessary. $timeLimit_2$ represents the maximum computational time in a subproblem or a cluster-problem after a feasible solution is found. This parameter value is lower with clustering applied, as the consumption regions are split into cluster-problems, which are solved using the RFFO, as explained in Subsection 5.2.2. With each cluster-problem being less complex, the need for computational time per cluster-problem is lower. The number of clusters to be made is set to $\lfloor \frac{\mathcal{J}^C}{3} \rfloor$, which ensures that all clusters contain at least 3 consumption ports. The clustering is therefore applied when $|\mathcal{J}^C| \geq 6$, since six is the smallest number of ports required to create at least two clusters.

Table 6.8: Initial value of matheuristic parameters.

MIPgap	10%
timeLimit₁	900 seconds
timeLimit₂	300 seconds (100 seconds with clustering)

6.4.1 Preprocessing

Four different combinations of the preprocessing techniques from Section 5.3 were tested. *No PP* is AD-RFFO with no preprocessing applied. *2nd Day* and *3rd Day* refers to only allowing departures every second and third day, respectively. *1D* refers to prohibiting each vessel from visiting one of the discharging ports in the region to which it belongs. Finally, *LA* is the preprocessing where all arcs with the longest travelling times in the region is removed. Keeping only departures in every third time period can be seen as a more aggressive preprocessing method than keeping every second, in terms of the number of variables removed, and combining either two with 1*D* or *LA* removes even more variables. Combinations including *LA* are regarded as the most aggressive. The preprocessing combinations presented have a different degree of aggressiveness to avoid testing all combinations. To see how different levels of aggressiveness in the preprocessing affect performance, these combinations were tested on the 20 initial instances, and the 10 initial complex instances.

Table 6.9 reports the results from testing the preprocessing methods on the 20 initial instances. The exact FCNF-model found a feasible solution within 3600s for 16 instances. The average gaps between these 16 primal solutions and the primal solutions of the heuristic applying the different preprocessing techniques is presented (Avg. Obj. vs. FCNF). The gap is calculated as $\frac{|ObjVal - ObjFCNF|}{|ObjVal|}$, where $ObjVal$ is the objective value found with the current preprocessing technique and *ObjFCNF* is the best objective value found by the FCNF. The number of instances where a feasible solution is found within 3600 seconds is reported for each of the preprocessing techniques (No. of sol.), as well as the average computational time (Avg. time). It appears to be a minor trade-off between the average runtime and the best objective value. The more aggressive the preprocessing, the quicker it finds its best solution, but the worse the objective value. This may indicate that in some cases, good solutions are cut out of the solution space when the preprocessing is too aggressive. In some cases, removing too many variables may result in the removal of all feasible solutions. This may be the reason that $2ndDay, LA$ only managed to find solution on 19 of the 20 instances.

Table 6.9: Results from the preprocessing tests conducted on the 20 initial instances, 3600s time limit per instance.

	No PP	**2nd Day**	**2nd Day, 1D**	**3rd Day, 1D**	**2nd Day, LA**
Avg. Obj. vs. FCNC	1,91%	1,85%	2,69%	3,20%	1,94%
No. of sol.	20	20	20	20	19
Avg. time [s]	976	910	780	775	994

In Table 6.10, the results from the tests run on the 10 complex instances are presented. To evaluate the ability to find high quality solutions within reasonable time, the number of best and feasible solutions are reported. The results supports the hypothesis that removing the longest arcs is too aggressive and impairs the ability to find feasible solutions. From the results on the complex instances, where the differences in performance were larger, it was clear that *3rd Day, 1D* is the best preprocessing for the matheuristic. It is superior at finding the best solutions as well as being among the best at finding feasible solutions. It removes more arcs than *2nd Day, 1D* while still finding better objective values, which suggests that it is more scalable for larger instances.

Table 6.10: Results from the preprocessing tests conducted on the 10 initial complex instances, 7200s time limit per instance.

	No PP	**2nd Day**	**2nd Day, 1D**	**3rd Day, 1D**	**2nd Day, LA**
No. of best	1	2	2	6	0
No. of sol.	9	7	7	9	5

6.4.2 Clustering

As described in Section 5.2, two different methods of clustering, namely demand and distance based clustering, were implemented. As discussed, both methods were only tested on the 10 initial complex instances.

The results from the testing is reported in Table 6.11, with the average objective value, average computational time and the number of feasible solutions reported for all configurations. "No clustering" finds only one feasible solution, and AD-RFFO without clustering is thus left out of the analysis. A reason for this may be that the larger subproblems are solved exactly, instead of the smaller cluster-problems which are solved iteratively with RFFO. Average objective value and computational time is based on the 6 instances where both clustering methods found a feasible solution within the time limit. When comparing the two, the number of feasible solutions is the most important, as the purpose of the clustering is to reduce computational complexity when subproblems in the matheuristic grow too large for commercial solvers. With feasible solutions to 9 out of 10 instances and a slightly better average objective value and computational time, distance clustering is preferred for the matheuristic.

Table 6.11: Results of clustering methods tested on the 10 initial complex instances, 7200s time limit per instance.

	Distance	**Demand**	**No Clustering**
Avg. obj. value	27 425 445	27 565 118	-
Avg. time [s]	3 699	4 409	-
No. of sol.	9	7	1

6.4.3 MIP-gap and time limits

The termination tolerances for dual gap and runtime for AD-RFFO were both subject to tuning. The dual gap tolerance is called $MIPGap$, and as discussed in Chapter 5, the tolerance for runtime is split into two, namely $timeLimit_1$ and $timeLimit_2$. As discussed, these sets upper limits for the computational time allowed to find a first feasible solution, and for optimizing after the first feasible solution is found, respectively. Both tolerances for dual gap and computational time affect the total time of each iteration in the AD-RFFO. In most cases, a lower dual gap tolerance translates to a higher computational time in each iteration, within the bounds of the time limit.

When solving larger instances, the clustering splits the subproblems into cluster-problems, which are solved iteratively. As there are multiple iterations on the clustering-problems within a subproblem when clustering is applied, it is natural to assume that different tolerances for dual gap and computational times are ideal when this is applied. As a consequence of this assumption, the testing was first performed on the 20 initial instances to decide which gap and time limit to use on smaller instances, and then on the 10 initial complex instances to decide which combination to use on larger instances where clustering is applied. When the dual gap tolerance is increased, so is the time spent on each iteration if this tolerance is not reached. There is an interdependence between the gap and time limit, and hence a need to tune them simultaneously as opposed to the sequential tuning of more independent parameters.

As discussed earlier, a 10% *MIPGap* was set as the standard tolerance for AD-RFFO. In addition, $timeLimit_1$ of 900 seconds is never subject to tuning, and 300 seconds for $timeLimit_2$ was set as the standard without clustering. In the rest of this section, $timeLimit_2$ is the only time limit considered. These tolerances were therefore used as a base for the tuning done in this section. To explore how both increases and decreases of the parameters affected performance, all combinations of 120, 300 and 480 seconds for the time limit and 5, 10 and 15% for the dual gap tolerance were tested. The results from tests run on the 20 initial instances can be seen in table Table 6.12. Average objective values and computational times for all 20 instances are reported, and every combination of parameters found feasible solutions to all instances. In addition, the average gaps to the objective values found by the FCNF model on 16 instances are reported.

Table 6.12: Results from combinations of MIP-gaps and time-limits on the 20 initial instances, 3600s time limit per instance.

Configuration	Avg. Obj. Value	Avg. Time [s]	Avg. Obj. vs. FCNF
5%, 120s	5 807 768	677	1,08%
5%, 300s	5 794 793	1288	0,88%
5%, 480s	5 792 095	1641	0,82%
10%, 120s	5 938 998	294	3,29%
10%, 300s	5 960 185	775	3,20%
10%, 480s	5 969 722	1004	3,20%
15%, 120s	6 115 788	160	5,11%
15%, 300s	6 117 727	264	5,17%
15%, 480s	6 096 039	474	5,17%

Some of the configurations allow for longer computational times in each subproblem, such as 5% gap and 480 seconds time limit. When gap limits are higher and time limits shorter, subproblems usually terminate faster. As the total computational time for an instance is constant, a faster termination of each subproblem allows for a higher number of iterations. The results suggest that increasing the gap limit worsens the objective value, which may indicate that finding better solutions for each subproblem in each iteration is more important for solution quality than the number of iterations. For the combinations with 10% and 15% tolerance, the objective value is almost constant for all time limits, although the average computational time increases significantly. This indicates that allowing computational times of more than 120 seconds increases solution times without improving the objective value significantly with gap limits of 10 and 15%.

As the aim of the matheuristic is finding high quality solution within two hours, a dual gap tolerance of 5% provides the best solution quality with acceptable computational times. As the difference in objective value between time limits of 300 and 480 seconds is marginal, 300 seconds was the preferred one. Consequently, the combination of 5% and 300s is the chosen combination when clustering is not applied.

For the 10 initial complex instances, where clustering is applied, a 10% tolerance for the dual gap and a 100s time limit was used initially. Similarly, to the testing conducted on the 20 initial instances, the parameters were in this case also increased and decreased, forming the combinations seen in Table 6.13. As in Table 6.12, the average objective values of the 5 instances solved by all combinations are reported, as well as the number of feasible solutions within the time limit of 7200 seconds. The average objective values of the combinations with a 40s time limit per iteration indicate a need for more time to have sufficiently good solution quality. A similar effect is seen in the low number of instances to which a feasible solution is found with this time limit. The solution times were quite similar for all instances, and thus the number of feasible solutions was the main emphasis when choosing the combination to use when clustering is applied. The 10% $MIPGap$ and 160s $timeLimit_2$ stands out as the best with 9 feasible solutions and slightly better average objective value than with similar gap and 100 seconds time limit.

Table 6.13: Results from combinations of MIP-gaps and time-limits on the 10 initial complex instances, 7200s time limit per instance.

Configuration	Avg. Obj. Value	No. of sol.
5%, 40s	48 923 892	6
5%, 100s	30 966 501	8
5%, 160s	30 854 371	6
10%, 40s	49 281 239	6
10%, 100s	32 504 557	9
10%, 160s	31 996 513	9
15%, 40s	49 687 995	6
15%, 100s	33 294 842	8
15%, 160s	33 031 713	8

6.4.4 Final Configuration of AD-RFFO

As a conclusion to the initial testing of the exact models and the tuning of AD-RFFO, a final configuration of AD-RFFO is presented in Table 6.14. The mathematical model and its configuration is presented, in addition to the combination of preprocessing methods, clustering method and termination tolerances. This configuration will be tested against the FCNF-model with valid inequality group FV1 applied in Section 6.5.

Table 6.14: The final configuration of AD-RFFO.

Mathematical Model	FCNF
Model Configuration	FV1
Preprocessing Method	3rd Day, 1D
Clustering Method	Distance
$\boldsymbol{MIPGap}$, $\boldsymbol{timeLimit_2}$ **without Clustering**	5%, 300s
$\boldsymbol{MIPGap}$, $\boldsymbol{timeLimit_2}$ **with Clustering**	10%, 160s

6.5 FCNF vs. AD-RFFO

A set of 75 new instances were generated to conduct the final testing. The results from this testing using FCNF and AD-RFFO is presented and discussed in this section. An overview of the main findings is given in Subsection 6.5.1, before the results of each instance group is analysed to find the effects of changes in instance parameters in Subsection 6.5.2.

6.5.1 Overview

The number of instances solved by the two different solution methods, as well as the number of best objective values, are given in Table 6.15. All the results for the instances can be seen in Appendix D.1. The dual gaps presented in the tables in this chapter were calculated as $\frac{|ObjVal-ObjBound|}{|ObjVal|}$, where *ObjBound* is the objective bound and *ObjVal* is the best objective value obtained before the time limit is reached. The objective bounds obtained by FCNF is used when calculating the dual gaps for AD-RFFO as well. The average dual gap of both solution methods, and the average difference between their objective values are also presented in Table 6.15. The difference in objective value is calculated by taking the average of AD-RFFO's objective values divided by FCNF's average objective value.

Table 6.15: Overview of the results from FCNF and AD-RFFO tested on all 75 instances. Time limit of 7200s on all instances.

	FCNF	**AD-RFFO**
No. of sol. (of 75)	37	75
No. of best*	27	10
Avg. dual gap*	7,25%	8,12%
Avg. obj. vs. FCNF	-	0,9%

*Only including instances both methods found feasible solutions

As can be seen in Table 6.15, FCNF manages to find feasible solutions only for about half of the instances, while AD-RFFO provides solutions on all test instances. Both models were allowed to run for 7200 seconds on every instance. The average dual gaps are relatively close between the two methods, and FCNF did not manage to prove optimality for any instances. The solutions found by AD-RFFO are close to the FCNF solutions, and in almost 30% of the instances solved by both methods, AD-RFFO finds better objective values. This indicates its capability to solve complex instances within reasonable time, and provide solutions that are very close to FCNF, and in some cases, even better.

The main reason AD-RFFO outperforms FCNF when solving large instances is that it takes advantage of dividing the full problem into subproblems. In this way, the number of variables constituting each subproblem are substantially reduced compared to the whole problem. Obtaining the full solution in this manner is easier than to solve the whole problem with all variables at the same time, as FCNF does. Additionally, the total number of variables is also reduced when using AD-RFFO as a result of the preprocessing and clustering heuristic applied, especially the integer variables. On average, AD-RFFO generated almost 70% fewer binary variables than FCNF. Only the binary variables are included in the analysis, as these contribute the most to computational complexity. The full overview with number of variables for each instance is given in Appendix D.2. Note that the variables reported are the variables after presolve in Gurobi. As will be seen in the following section, AD-RFFO is less affected by the increases in the different parameters in the instances than FCNF, mainly because of the problem decomposition and the reduction in the number of variables.

6.5.2 Parameter Impact on Solution Methods

Studying the different instance parameters, it can be seen that changes in some parameters affect the solutions more than others. Instance groups that exhibit clear patterns in terms of how an increase in their respective parameter affect the solutions are discussed in this section. Tables presented in this section include the relevant instances that are discussed (Instance), the dual gaps for FCNF and AD-RFFO (Dual Gap), and the solution times for AD-RFFO (Solution Time).

Time Group

As the number of time periods in the instances increase, it becomes more difficult for FCNF to find feasible solutions. This is reflected in the duality gaps, which are increasing as the instance size grow, as seen in Table 6.16. FCNF only manages to find feasible solutions on the five smallest instances in this group. AD-RFFO does not exhibit the same pattern of increased difficulty to solve the problem as time periods increase. Feasible solutions are found for every instance, and the objective values do not deviate significantly from the exact method. Even when solving the larger instances, the computational time does not increase substantially.

Table 6.16: Results from both FCNF and AD-RFFO in the group with an increase in planning periods.

	FCNF	**AD-RFFO**	
Instance	Dual Gap	Solution Time [s]	Dual Gap
T20	2,73%	540	3,93%
T40	8,68%	5400	9,18%
T60	9,85%	1260	10,03%
T80	7,29%	1020	9,12%
T100	12,40%	660	8,36%
T120	-	728	-
T140	-	3680	-
T160	-	1821	7,85%
T180	-	4518	-
T200	-	1060	-

Another aspect that was tested is variation in demand. While keeping all other parameters the same as in instance group *Time*, the demand was changed to be variable during the planning horizon. There was no clear impact of variable demand on the solution methods. The same pattern as in the time group was seen as the instances grew.

Region Group

Studying the results from the instance group with an increase in the number of regions, it can be seen in Table 6.17 that all the instances up to R10 are solved by both solution methods, but feasible solutions for larger instances than R10 are only found by AD-RFFO. For the three largest instances, the computational nodes ran out of memory during the creation of the model when FCNF was applied. There is no prominent pattern of how an increase in the number of regions affect AD-RFFO other than a higher average solution time on the larger instances compared to the smaller once. This seems reasonable because as the number of regions increase, the number of subproblems to be solved increases as well. As the subproblems are solved independently, solving the full problem itself does not become more difficult. This can also be seen looking at the number of binary variables in each subproblem, given in Figure 6.1. The average numbers of binary variables per subproblem are relatively constant for all instances, but the total number of variables increases.

Table 6.17: Results from both FCNF and AD-RFFO in the group with an increase in regions.

	FCNF	**AD-RFFO**	
Instance	Dual Gap	Solution Time[s]	Dual Gap
R2	4,05%	902	10,84%
R3	8,68%	5400	9,18%
R4	3,83%	637	6,51%
R5	6,90%	1219	8,08%
R6	5,53%	3454	8,03%
R7	6,42%	6000	7,87%
R8	7,26%	5640	8,67%
R9	5,95%	1782	6,81%
R10	5,10%	1929	6,73%
R11	-	3733	7,42%
R12	-	6242	8,00%
R15	-	2780	8,30%
R20	-*	6832	-**
R25	-*	6075	-**
R30	-*	7623	-**

*The nodes ran out of memory to create model and start computation

**No dual bound found for FCNF, AD-RFFO found solution

The advantage of decomposing the problem into subproblems becomes greater as the number of regions increase. As can be seen in Figure 6.1, the number of variables increase with the number of regions. For FCNF, all the variables are included simultaneously when optimizing the whole problem. When using AD-RFFO, the preprocessing initially makes sure that the total variables constituting the problem are substantially reduced. A reduction of at least 50% is made for each instance. Furthermore, the advantage of using AD-RFFO is seen as only one subproblem is solved at a time, meaning that each optimization done in AD-RFFO only considers a fraction of the variables of FCNF. Even though the problems are decomposed and the number of variables are reduced, it appears that the solutions provided by AD-RFFO are close to the objective bounds found by FCNF regardless of the number of regions.

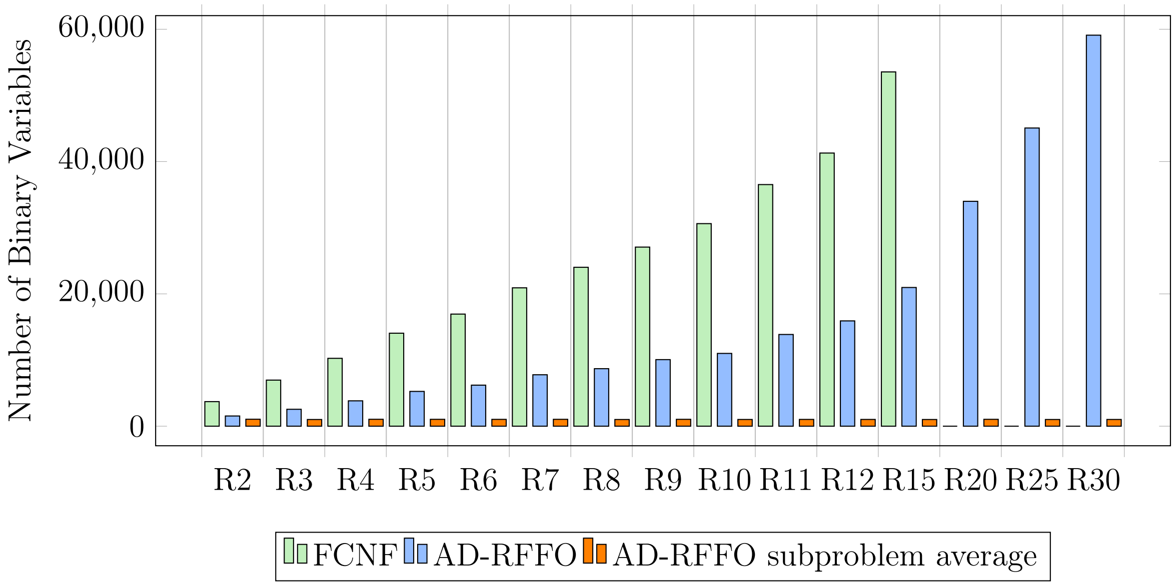

Figure 6.1: Number of binary variables in the region group for FCNF, AD-RFFO and the average for subproblems of AD-RFFO.

Port Group

An increase in the number of ports in each consumption region increases the number of arcs, and thus binary variables, drastically. Consequently, the problem complexity increases, which can be seen in Table 6.18.

Table 6.18: Results from both FCNF and AD-RFFO in the group with an increase in consumption ports.

	FCNF	**AD-RFFO**	
Instance	Dual Gap	Solution Time [s]	Dual Gap
J1	3,10%	7	3,62%
J2	2,53%	607	3,25%
J3	8,68%	5400	9,18%
J4	8,72%	5400	8,17%
J5	14,89%	3048	12,74%
J6	21,56%	5793	19,33%
J7	-	5825	23,68%
J8	-	6069	22,53%

The dual gaps increase for both solution methods as the number of ports increase. Increasing the number of ports also affects the computational time of AD-RFFO, which gets higher as the instances get larger. One of the main reasons for this is that as the number of ports get larger, and the number of vessels is kept constant, the routing within each region may become more complex as each vessel must be routed through more ports than previously. An indication of this will be seen when looking at the vessel group, where an increase in number of vessels allows the matheuristic to obtain feasible solutions faster. The increased difficulty to solve the problem with more ports is also reflected in the number of binary variables that are generated when solving the instances, which rapidly increases with the number of ports, as seen in the plot in Figure 6.2.

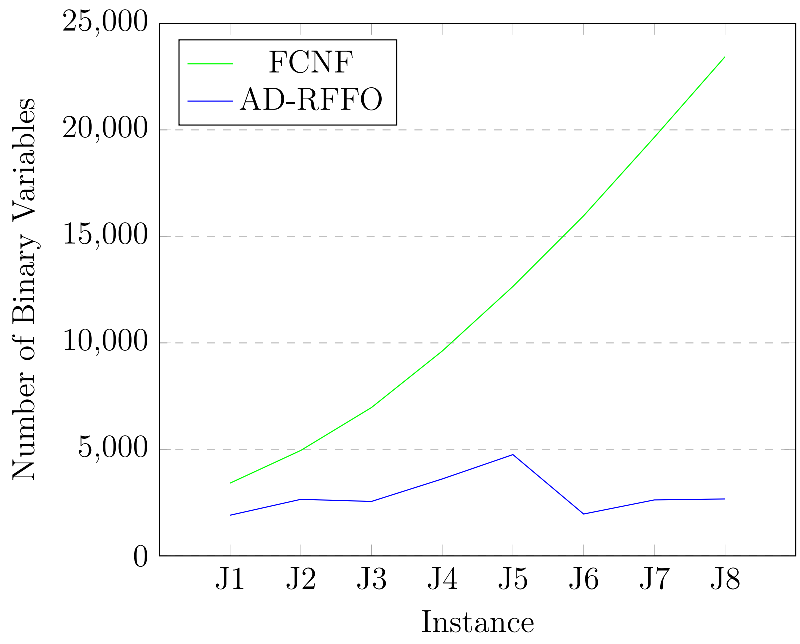

Figure 6.2: Number of binary variables in the port group for FCNF and AD-RFFO.

For FCNF, the number of variables has an increasing growth as the number of ports increases, which is illustrated by the convex plot in Figure 6.2. This is not the case for AD-RFFO. It exhibits a marginal growth in the number of variables until it reaches instance J6, where the number decreases, and does not increase significantly in the following instances. When the number of ports increases above five, the clustering is applied to each subproblem. Clustering reduces the number of variables significantly, and creates even smaller cluster-problems that are solved efficiently. This allows AD-RFFO to find feasible solutions for every instance, including the two where FCNF did not. By solving more instances and obtaining better solutions on three of the six instances solved by FCNF, AD-RFFO seems to scale better when the number of ports increase.

Vessel Group

As the number of vessels increase, an opposite effect from the increase in number of ports can be seen. Whereas an increase in the number of ports worsened duality gaps and computational times, both are improved by having more vessels in each region. Feasible solutions are found for all instances, as seen in Table 6.19. When examining the computational time of AD-RFFO, a reduction is evident when the number of vessels increases. This reflects that the problems are getting less difficult to solve as the instances grow. More available vessels creates flexibility in terms of routing, and each vessel has the possibility to visit fewer customers to obtain a feasible solution compared to when having fewer vessels per port. This makes the routing part of the problem easier to solve. The reduction in complexity by having more vessels is also shown by FCNF obtaining better duality gaps than AD-RFFO on all the instances. Using a heuristic, certain simplifications are made to reduce complexity. For smaller problems, these simplifications might not be beneficial for obtaining good solutions compared to solving the problem exactly. The preprocessing used in the heuristic might also be unfavourable, especially for smaller problems, by removing good solutions from the solution space. Therefore, applying a matheuristic on these instances might speed up the calculation, at the expense of objective value.

Table 6.19: Results from both FCNF and AD-RFFO in the group with an increase in the number of vessels.

	FCNF	**AD-RFFO**	
Instance	Dual Gap	Solution Time [s]	Dual Gap
V1	2,78%	4920	6,02%
V2	9,34%	3000	10,15%
V3	8,68%	5400	9,18%
V4	2,74%	1200	6,27%
V5	3,35%	973	7,19%
V6	2,79%	37	5,52%
V7	1,66%	54	6,94%
V8	1,97%	52	3,36%

Product Group

To test whether the solution methods were sensitive to changes in the number of products in the problem, a set of instances differing in number of products was tested. There was no clear impact of increasing the number of products found during the testing, and all instances with multiple products were solved with reasonable dual gaps by both solution methods. One of the reasons that an increase in the number of products did not make any impact on the solution methods might be that the number of binary variables generated in each instance remained constant, as can be seen in Figure 6.3. The reason for this being that no binary variables are connected to products. As long as the number of binary variables does not increase significantly, it seems like the problem does not increase substantially in the degree of difficulty. The assumption that each vessel has numerous tanks, and that combinations of tanks allow for every ratio of products, is the main reason for products not being connected to any binary variables. If this was not the case, a part of the problem would be tank allocation, which would have contributed to making the problems more difficult to solve as the number of products increased.

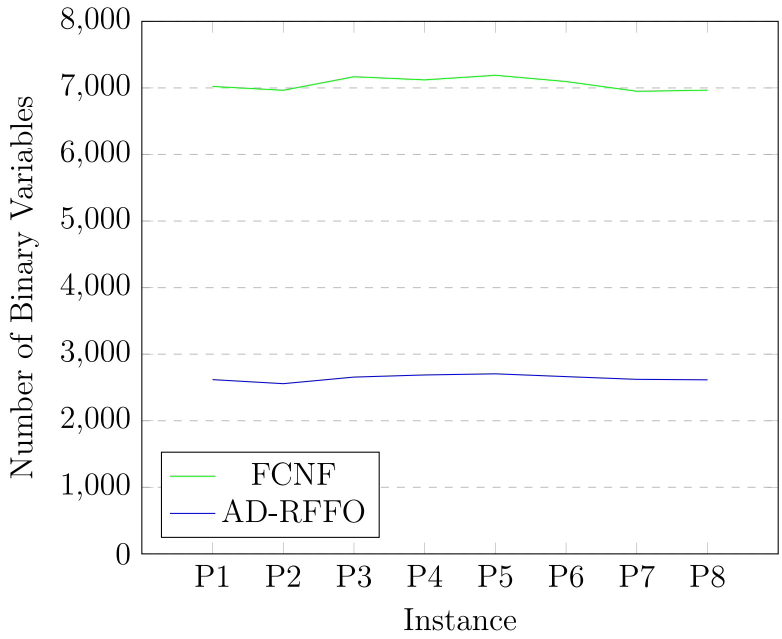

Figure 6.3: Number of variables in port group for FCNF and AD-RFFO.

6.5.3 Complex Group

The group of complex instances consists of instances varying in several dimensions. The purpose was measuring the performance of AD-RFFO when applied to computationally heavy instances, and test whether or not the proposed matheuristic manages to solve instances which FCNF could not. All instances were solved, which can be seen in Table 6.20. Note that only the relevant parameters changed are presented, with all parameters of the instances presented in Appendix C.1. FCNF did not solve any of the instances within the time limit, and managed only to obtain bounds on three of the instances. The solutions obtained by AD-RFFO had an average gap of 20% to the three bounds obtained, but the quality of these bounds is uncertain.

Table 6.20: Results from both FCNF and AD-RFFO in the group with complex instances.

	AD-RFFO			
Instance	$\|\mathcal{T}\|$	$\|\mathcal{R}\|$	$\|\mathcal{J}\|$	Solution Time [s]
C1.1	40	4	6	5856
C1.2	60	4	6	6437
C1.3	80	4	6	5867
C1.4	100	4	6	7110
C1.5	40	4	7	7039
C1.6	60	4	7	7948
C1.7	80	4	7	6233
C1.8	40	4	8	7867
C1.9	60	4	8	6000
C1.10	80	4	8	8073
C2.1	50	5	6	1039
C2.2	100	5	6	6360
C2.3	100	5	6	7831
C2.4	50	5	7	6872
C2.5	100	5	7	4190
C2.6	50	5	8	7094
C2.7	100	5	8	5220
C2.8	50	5	9	6254
C2.9	100	5	9	5672
C2.10	40	5	12	3725

As mentioned, AD-RFFO takes advantage of the preprocessing and problem decomposition to obtain subproblems with fewer variables. In addition to a lower number of variables for the total problem, AD-RFFO has an advantage by only solving the subproblems instead of the full problem. The number of variables generated is one of the main attributes that distinguishes FCNF and AD-RFFO. When looking at solutions of the previous instance groups, it was clear that AD-RFFO generated fewer variables than FCNF. When solving larger instances, the difference between these two methods becomes more prominent. Looking at Figure 6.4, it can be seen that AD-RFFO generates between 82%-95% fewer binary variables for every instance compared to FCNF.

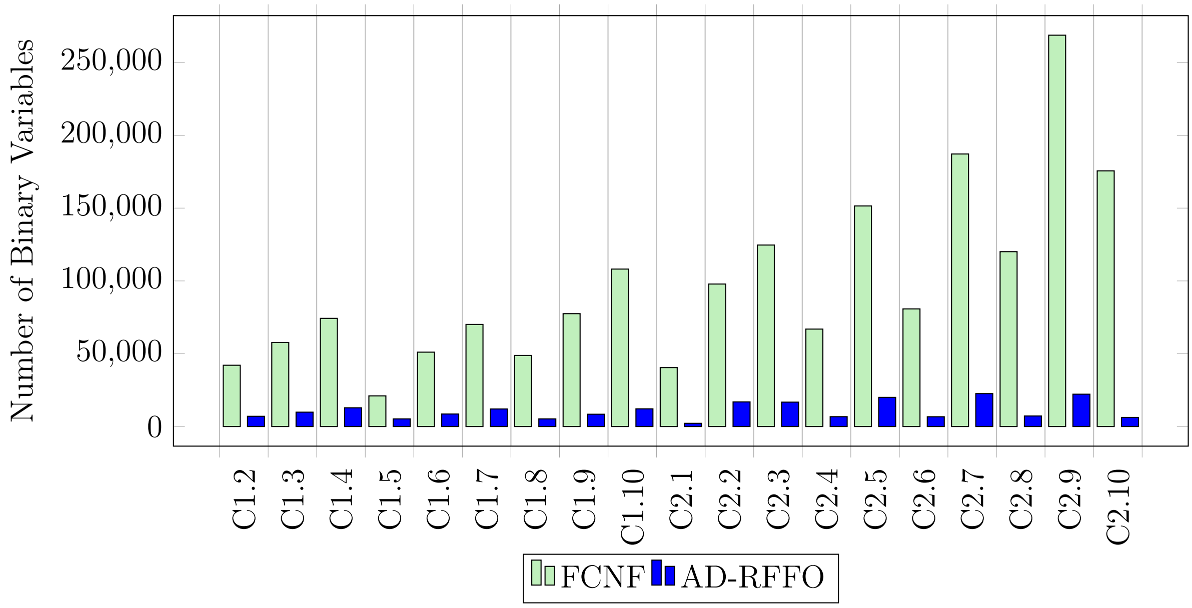

Figure 6.4: Number of variables in the complex instances for FCNF and AD-RFFO.

AD-RFFO finds feasible solutions to all complex instances, and drastically reduces the number of variables in each subproblem. Based on these results, it can be assumed that even more complex instances, especially in the number of regions and time periods, could be solved with satisfactory solution quality.

Chapter 7

Concluding Remarks

In this master's thesis, a maritime inventory routing problem inspired by maritime supply chains with intermediate facilities and multiple products is studied. To the best of our knowledge, MIRPs with two echelons have not previously been studied. With the combination of two-echelons and multiple products, the problem stands out from other problems in the MIRP literature. We present two mathematical models for this problem, where one of the models is a general arc-flow model, and the other being an reformulation of this model to a fixed-charged network flow (FCNF) model. A decomposition matheuristic is constructed and applied to the problem, with an additional cluster-based Relax-and-Fix and Fix-and-Optimize heuristic for solving the decomposed subproblems. In addition, preprocessing techniques are presented and applied to reduce the total problem size for the matheuristic. The matheuristic demonstrates how the two-echelon problem studied can be decomposed into subproblems based on regions.

75 problem instances were generated for the computational study, with parameters and values based on real life data. Instances differ in the number of regions, ports per consumption region, vessels per region, products and time periods. An subset of all instances were used for initial testing. The FCNF model proved to be the best suited for use in both the matheuristic and final testing against the matheuristic. The FNCF model found more feasible solutions, and obtained better dual bounds and objective values after both ten minutes and one hour, compared to the general arc-flow model.

The matheuristic finds feasible solutions for all 75 instances tested, compared to the exact solution method's 37 feasible solutions. Of these 37 instances, the solutions of the matheuristic deviate no more than 0.9% from FCNF's objective values. The simplifications and reduction of problem size applied by the matheuristic reduce the quality of the solutions slightly, but is significantly more scalable for large instances than the FCNF model.

The study conducted to test the effect of changes in different parameters of test instances can help explain the results of the heuristic. When increasing the number of regions, the effect is small on the performance of the AD-RFFO. By decomposing the full problem, this increase only results in more subproblems to solve sequentially, without increasing the complexity of the entire problem. When increasing the number of ports per region, the effect of the cluster-based R&F/F&O heuristic applied to the subproblems is evident. By both reducing the number of variables and solving smaller cluster-problems iteratively, the AD-RFFO can solve problems with significantly higher number of ports than FCNF. The ability of the AD-RFFO to solve large problem instances is a result of combining these two heuristic methods with preprocessing.

The purpose of this thesis was to implement a model for solving complex instances of the two-echelon multi-product maritime inventory routing problem, and develop a matheuristic that is able to solve large instances and obtain high-quality solutions. The matheuristic has demonstrated how the problem of this report can be decomposed and solved iteratively, and has exhibited promising results in terms of the problem sizes solved, and the quality of the solutions compared to the FCNF model. Thus, a basis for future research of both exact and heuristic solution methods for two echelon multi-product maritime inventory routing problems has been established.

Chapter 8

Future Research

This master's thesis contains several parts subject to future research. The problem and the corresponding mathematical models, respectively presented in Chapter 3 and Chapter 4, all include assumptions and limitations. Many of these assumptions and limitations are not present in real life, and an interesting field of study in future research is therefore the extension of this model to reflect real life in a better way. In this thesis, it is for instance assumed that the hubs can buy and sell an unlimited amount of products to and from the spot market with instant delivery. In real life, bounds on the amount bought and sold, and delivery time of products bought, may be appropriate. Other potential extensions include chartering of vessels, the allocation of tanks to vessels, and more realistic loading and discharging in ports, such as restricting the number of vessels allowed to load or discharge a specific product simultaneously.

The matheuristic constructed for the problem in this thesis applies a cluster-based R&F/F&O heuristic for the smaller subproblems when the full problem is decomposed. As each subproblem can be solved as one-echelon MIRPs, there are a vast number of possible solution methods and heuristics in the literature that may be applied to these smaller subproblems. Future work on the matheuristic includes the testing of different solution methods for each subproblem, in addition to other preprocessing and decomposition heuristics for the full problem.

Furthermore, finding better estimates of the performance of the matheuristic is an important part of future research. This includes research with the purpose of finding good dual bounds for the matheuristic's solutions. New mathematical models and other techniques can be implemented with this purpose.

Bibliography

Agra, A., Andersson, H., Christiansen, M. and Wolsey, L. (2013), 'A maritime inventory routing problem: Discrete time formulations and valid inequalities', *Networks* **62**(4), 297–314.

Agra, A., Christiansen, M., Delgado, A. and Simonetti, L. (2014), 'Hybrid heuristics for a short sea inventory routing problem', *European Journal of Operational Research* **236**, 924–935.

Al-Khayyal, F. and Hwang, S.-J. (2007), 'Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part i: Applications and model', *European Journal of Operational Research* **176**(1), 106–130.

Allianz (2021), Safety and shipping review 2021 - an annual review of trends and developments in shipping losses and safety, Technical report, Allianz.

Andersson, H. (2011), 'A maritime pulp distribution problem', *INFOR: Information Systems and Operational Research* **49**(2), 125–138.

Archetti, C. and Speranza, M. (2014), 'A survey on matheuristics for routing problems', *EURO Journal on Computational Optimization* **2**(4), 223–246.

Asokan, B. V. (2014), Parallel large-neighborhood search techniques for lng inventory routing.

Bertazzi, L. and Speranza, G. (2012), 'Inventory routing problems: an introduction', *EURO Journal on Transportation and Logistics* **176**(1), 307–326.

Bilgen, B. and Ozkarahan, I. (2007), 'A mixed-integer linear programming model for bulk grain blending and shipping', *International Journal of Production Economics* **107**(2), 555–571. Operations Management in China.

Chan, L. M. A. and Simchi-Levi, D. (1998), 'Probabilistic analyses and algorithms for three-level distribution systems', *Management Science* **44**(11), 1562–1576.

Chaudhary, F., Fossum, E. and Drangsland, H. (2021), 'A general model and a heuristic for two-echelon multi-product maritime inventory routing problems', - **1**(1).

Christiansen, M. and Fagerholt, K. (2009), *Maritime inventory routing problems*, Springer US, Boston, MA, pp. 1947–1955.

Christiansen, M., Fagerholt, K., Flatberg, T., Øyvind Haugen, Kloster, O. and Lund, E. H. (2011), 'Maritime inventory routing with multiple products: A case study from the cement industry', *European Journal of Operational Research* **208**(1), 86–94.

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

Christiansen, M. and Grønhaug, R. (2009), *Supply Chain Optimization for the Liquefied Natural Gas Business*, Springer Verlag Berlin Heidelber, Boston, MA, pp. 195–218.

Cuda, R., Guastaroba, G. and Speranza, M. (2015), 'A survey on two-echelon routing problems', *Computers Operations Research* **55**(1), 185–199.

EEA, E. (2021), European maritime transport environmental report 2021, Technical report, European Environment Agency.

Engineer, F. G., Furman, K. C., Nemhauser, G. L., Savelsbergh, M. W. P. and Song, J.-H. (2012), 'A branch-price-and-cut algorithm for single-product maritime inventory routing', *Operations Research* **60**(1), 106–122.

Fagerholt, K. and Christiansen, M. (2000), 'A combined ship scheduling and allocation problem', *Journal of the Operational Research Society* **51**(7), 834–842.

Farias, K., Hadj-Hamou, K. and Yugma, C. (2021), 'Model and exact solution for a two-echelon inventory routing problem', *International Journal of Production Research* **59**(10), 3109–3132.

Ferreira, D., Morabito, R. and Rangel, S. (2009), 'Solution approaches for the soft drink integrated production lot sizing and scheduling problem', *European Journal of Operational Research* **196**, 697–706.

Fischetti, M. and Fischetti, M. (2016), *Matheuristics*, Springer International Publishing, Cham, pp. 1–33.

Friske, M. W. and Buriol, L. S. (2017), A relax-and-fix algorithm for a maritime inventory routing problem, *in* T. Bektaş, S. Coniglio, A. Martinez-Sykora and S. Voß, eds, 'Computational Logistics', Springer International Publishing, Cham, pp. 270–284.

Friske, M. W. and Buriol, L. S. (2018), Applying a relax-and-fix approach to a fixed charge network flow model of a maritime inventory routing problem, *in* R. Cerulli, A. Raiconi and S. Voß, eds, 'Computational Logistics', Springer International Publishing, Cham, pp. 3–16.

Friske, M. W., Buriol, L. S. and Camponogara, E. (2021), 'A relax-and-fix and fix-and-optimize algorithm for a maritime inventory routing problem', *Computers Operations Research* **137**, 105520.

Gaur, V. and Fisher, M. L. (2004), 'A periodic inventory routing problem at a supermarket chain', *Operations Research* **52**(6), 813–822.

Goel, V., Furman, K. C., Song, J.-H. and El-bakry, A. S. (2012), 'Large neighborhood search for lng inventory routing', *Journal of Heuristics* **18**.

Grønhaug, R., Christiansen, M., Desaulniers, G. and Desrosiers, J. (2010), 'A branch-and-price method for a liquefied natural gas inventory routing problem', *Transportation Science* **44**(3), 400–415.

Guimarães, T. A., Coelho, L. C., Schenekemberg, C. M. and Scarpin, C. T. (2019), 'The two-echelon multi-depot inventory-routing problem', *Computers Operations Research* **101**(1), 220–233.

Hemmati, A., Hvattum, L. M., Christiansen, M. and Laporte, G. (2016), 'An iterative two-phase hybrid matheuristic for a multi-product short sea inventory-routing problem', *European Journal of Operational Research* **252**(3), 775–788.

Hemmati, A., Stålhane, M., Hvattum, L. M. and Andersson, H. (2015), 'An effective heuristic for solving a combined cargo and inventory routing problem in tramp shipping', *Computers Operations Research* **64**, 274–282.

Hennig, F., Nygreen, B., Christiansen, M., Fagerholt, K., Furman, K., Song, J., Kocis, G. and Warrick, P. (2012), 'Maritime crude oil transportation – a split pickup and split delivery problem', *European Journal of Operational Research* **218**(3), 764–774.

Li, J., Chu, F. and Chen, H. (2011), 'A solution approach to the inventory routing problem in a three-level distribution system', *European Journal of Operational Research* **210**(3), 736–744.

Nambirajana, R., Mendoza, A., Pazhani, S., Narendran, T. and Ganesh, K. (2016), 'Care: Heuristics for two-stage multi-product inventory routing problems with replenishments', *Computers Industrial Engineering* **97**, 41–57.

Papageorgiou, D. J., Cheon, M.-S., Harwood, S., Trespalacios, F. and Nemhauser, G. L. (2018), *Recent Progress Using Matheuristics for Strategic Maritime Inventory Routing*, Springer International Publishing, Cham, pp. 59–94.

Papageorgiou, D. J., Nemhauser, G. L., Sokol, J., Cheon, M.-S. and Keha, A. B. (2014a), 'Mirplib – a library of maritime inventory routing problem instances: Survey, core model, and benchmark results', *European Journal of Operational Research* **235**(2), 350–366. Maritime Logistics.

Papageorgiou, D. J., Nemhauser, G. L., Sokol, J. and Keha, A. (2014b), 'Two-stage decomposition algorithms for single product maritime inventory routing', *INFORMS* **26**(4), 825–847.

Persson, J. A. and Göthe-Lundgren, M. (2005), 'Shipment planning at oil refineries using column generation and valid inequalities', *European Journal of Operational Research* **163**(3), 631–652. Supply Chain Management and Advanced Planning.

Rakke, J. G., Stålhane, M., Moe, C. R., Christiansen, M., Andersson, H., Fagerholt, K. and Norstad, I. (2011), 'A rolling horizon heuristic for creating a liquefied natural gas annual delivery program', *Transportation Research Part C: Emerging Technologies* **19**(5), 896–911. Freight Transportation and Logistics (selected papers from ODYSSEUS 2009 - the 4th International Workshop on Freight Transportation and Logistics).

Rohmer, S., Claassen, G. and Laporte, G. (2019), 'A two-echelon inventory routing problem for perishable products', *Computers Operations Research* **107**, 156–172.

Ronen, D. (2002), 'Marine inventory routing: shipments planning', *Journal of the Operational Research Society* **53**(1), 108–114.

Sanghikian, N., Martinelli, R. and Abu-Marrul, V. (2021), A hybrid vns for the multi-product maritime inventory routing problem, *in* N. Mladenovic, A. Sleptchenko, A. Sifaleras and M. Omar, eds, 'Variable Neighborhood Search', Springer International Publishing, Cham, pp. 111–122.

ShipTraffic (2022), 'Sea distance calculator', http://www.shiptraffic.net/2001/05/sea-distances-calculator.html.

Sirimanne, S. N., Hoffman, J., Juan, W., Asariotis, R., Assaf, M., Ayala, G., Benamara, H., Chantrel, D., Hoffmann, J., Premti, A. et al. (2019), Review of maritime transport 2019, Technical report, tech. rep.

Siswanto, N., Essam, D. and Sarker, R. (2011), 'Solving the ship inventory routing and scheduling problem with undedicated compartments', *Computers Industrial Engineering* **61**, 289–299.

Song, J.-H. and Furman, K. C. (2013), 'A maritime inventory routing problem: Practical approach', *Computers Operations Research* **40**(3), 657–665. Transport Scheduling.

Soroush, H. and Al-Yakoob, S. (2018), 'A maritime scheduling transportation-inventory problem with normally distributed demands and fully loaded/unloaded vessels', *Applied Mathematical Modelling* **53**, 540–566.

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

Tankers (2014), 'Tankers', Available at http://hb.hr/wp-content/uploads/2014/12/tankers.pdf (2022/02/07).

Uggen, K. T., Fodstad, M. and Nørstebø, V. S. (2013), 'Using and extending fix-and-relax to solve maritime inventory routing problems', *TOP - An Official Journal of the Spanish Society of Statistics and Operations Research* **21**(2), 355–277.

United Nations (2016), 'Maritime transport is 'backbone of global trade and the global economy', says secretary-general in message for international day', Available at https://www.un.org/press/en/2016/sgsm18129.doc.htm (2021/07/12).

Vadseth, S. T., Andersson, H. and Stålhane, M. (2021), 'An iterative matheuristic for the inventory routing problem', *Computers Operations Research* **131**, 105262.

Yieldstreet (2019), 'The most common types of large cargo ships, explained', Available at https://www.yieldstreet.com/resources/article/types-of-cargo-ships/ (2022/02/05).

Zhao, Q.-H., Chen, S. and Zang, C.-X. (2008), 'Model and algorithm for inventory/routing decision in a three-echelon logistics system', *European Journal of Operational Research* **191**(3), 623–635.

Appendices

Appendix A

Mathematical Models

A.1 Compact Basic Arc-Flow Model

$$\min \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}_v} C_{va} x_{va} + \sum_{(j,t) \in \mathcal{N}^{\mathcal{H}}} \sum_{p \in \mathcal{P}} (C^B k_{jtp} - C^S l_{jtp}), \tag{A.1}$$

$$\sum_{a \in \delta^+_{vn}} x_{va} - \sum_{a \in \delta^-_{vn}} x_{va} = \begin{cases} 1 & \text{if } n = n_0 \\ -1 & \text{if } n = n_{(T)} \\ 0 & \text{if } n \in \mathcal{N} \end{cases} \quad v \in \mathcal{V}, n \in \mathcal{N}_0 \tag{A.2}$$

$$z_{jvt} \leq \sum_{a \in \delta^-_{vn}} x_{va}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, \tag{A.3}$$

$$z_{jvt} \leq x_{va}, \quad v \in \mathcal{V}, (j,t) \in \mathcal{N}, a = \{(j,t),(j,t+1)\}, \tag{A.4}$$

$$z_{jv(t-1)} \leq z_{jvt} + \sum_{a = ((j,t),(i,t')) \in \delta^+_{vn}, i \neq j} x_{va}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, t > 0 \tag{A.5}$$

$$s_{jtp} = s_{j(t-1)p} + D_{jtp} - \sum_{v \in \mathcal{V}_r} \Delta_j f_{jvtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{A.6}$$

$$s_{j0p} = S^0_{jp} + D_{j0p} - \sum_{v \in \mathcal{V}_r} \Delta_j f_{jv0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, p \in \mathcal{P}, \tag{A.7}$$

$$s_{jtp} = s_{j(t-1)p} + \sum_{v \in \mathcal{V}_0} f_{jvtp} - \sum_{v \in \mathcal{V}_r} f_{jvtp} + k_{jtp} - l_{jtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}^{\mathcal{H}} \cap \mathcal{J}_r, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{A.8}$$

$$s_{j0p} = S^0_{jp} + \sum_{v \in \mathcal{V}_0} f_{jv0p} - \sum_{v \in \mathcal{V}_r} f_{jv0p} + k_{j0p} - l_{j0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}^{\mathcal{H}} \cap \mathcal{J}_r, p \in \mathcal{P}, \tag{A.9}$$

$$s_{jtp} \geq (S^{max}_{jp} + S^{min}_{jp})/2, \quad n = (j,t) \in \mathcal{N}, p \in \mathcal{P}, t = |\mathcal{T}| - 1, \tag{A.10}$$

$$u_{vtp} = u_{v(t-1)p} + \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{P}}} f_{jvtp} - \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{H}}} f_{jvtp}, \quad v \in \mathcal{V}_0, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{A.11}$$

$$u_{v0p} = U^0_{vp} + \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{P}}} f_{jv0p} - \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{H}}} f_{jv0p}, \quad v \in \mathcal{V}_0, p \in \mathcal{P}, \tag{A.12}$$

$$u_{vtp} = u_{v(t-1)p} + \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{H}}} f_{jvtp} - \sum_{n=(j,t) \in \mathcal{N}^{\mathcal{C}}} f_{jvtp}, \quad v \in \mathcal{V} \setminus \{\mathcal{V}_0\}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{A.13}$$

$$u_{v0p} = U^0_{vp} + \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{H}}} f_{jv0p} - \sum_{n=(j,0) \in \mathcal{N}^{\mathcal{C}}} f_{jv0p}, \quad v \in \mathcal{V} \setminus \{\mathcal{V}_0\}, p \in \mathcal{P}, \tag{A.14}$$

$$\sum_{v \in \mathcal{V}} z_{jvt} \leq B_j, \quad n = (j,t) \in \mathcal{N}, \tag{A.15}$$

$$F^{min}_j z_{jvt} \leq \sum_{p \in \mathcal{P}} f_{jvtp} \leq F^{max}_j z_{jvt}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, \tag{A.16}$$

$$S^{min}_{jp} \leq s_{jtp} \leq S^{max}_{jp}, \quad n = (j,t) \in \mathcal{N}, p \in \mathcal{P}, \tag{A.17}$$

$$0 \leq \sum_{p \in \mathcal{P}} u_{vtp} \leq Q_v, \quad v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.18}$$

$$0 \leq k_{jtp}, l_{jtp}, \quad (j,t) \in \mathcal{N}^{\mathcal{H}}, p \in \mathcal{P}, \tag{A.19}$$

$$0 \leq s_{jtp} \quad (j,t) \in \mathcal{N}, p \in \mathcal{P}, \tag{A.20}$$

$$0 \leq f_{jvtp} \quad (j,t) \in \mathcal{N}, v \in \mathcal{V}, p \in \mathcal{P}, \tag{A.21}$$

$$0 \leq u_{vtp} \quad v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.22}$$

$$x_{va} \in \{0,1\}, \quad v \in \mathcal{V}, a \in \mathcal{A}_v, \tag{A.23}$$

$$z_{jvt} \in \{0,1\}, \quad v \in \mathcal{V}, n = (j,t) \in \mathcal{N}, \tag{A.24}$$

A.2 Compact FCNF Formulation

$$\min \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{J}} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}} C_{ijv} x_{ijvt} + \sum_{j \in \mathcal{J}^{\mathcal{H}}} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} (C^B k_{jtp} - C^S l_{jtp}), \tag{A.25}$$

$$\sum_{i \in \mathcal{J} \cup O_v} x_{ijv(t-T_{ijv})} + w_{jv(t-1)} = w_{jvt} + z^A_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \setminus \{0\}, \tag{A.26}$$

$$z^A_{jv(t-1)} + z^B_{jv(t-1)} = z^B_{jvt} + \sum_{i \in \mathcal{J} \cup D_v} x_{jivt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \setminus \{0\}, \tag{A.27}$$

$$\sum_{j \in \mathcal{J} \cup D_v} x_{O_v jv0} = 1, \quad v \in \mathcal{V}, \tag{A.28}$$

$$\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{J} \cup O_v} x_{iD_v vt} = 1, \quad v \in \mathcal{V}, \tag{A.29}$$

$$z^A_{jvt} + z^B_{jvt} = z_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.30}$$

$$\sum_{i \in \mathcal{J} \cup O_v} u^X_{ijv(t-T_{ijv})p} + u^W_{jv(t-1)p} = u^W_{jvtp} + u^A_{jvtp}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{A.31}$$

$$u^A_{jv(t-1)p} + u^B_{jv(t-1)p} + \Delta_j f_{jv(t-1)p} = u^B_{jvtp} + \sum_{i \in \mathcal{J} \cup D_v} u^X_{jivtp}, \quad j \in \mathcal{J} \setminus \{\mathcal{J}^{\mathcal{H}}\}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.32}$$

$$u^A_{jv(t-1)p} + u^B_{jv(t-1)p} + \Delta_v f_{jv(t-1)p} = u^B_{jvtp} + \sum_{i \in \mathcal{J} \cup D_v} u^X_{jivtp}, \quad j \in \mathcal{J}^{\mathcal{H}}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.33}$$

$$u^X_{O_v jv0p} = U^0_{vp} x_{O_v jv0}, \quad v \in \mathcal{V}, j \in \mathcal{J} \cup D_v, p \in \mathcal{P}, \tag{A.34}$$

$$\sum_{p \in \mathcal{P}} u^X_{ijvtp} \leq Q_v x_{ijvt}, \quad v \in \mathcal{V}, i \in \mathcal{J} \cup O_v, j \in \mathcal{J} \cup D_v, t \in \mathcal{T}, \tag{A.35}$$

$$\sum_{p \in \mathcal{P}} u^A_{jvtp} \leq Q_v z^A_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.36}$$

$$\sum_{p \in \mathcal{P}} u^B_{jvtp} \leq Q_v z^B_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.37}$$

$$\sum_{p \in \mathcal{P}} u^W_{jvt} \leq Q_v w_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.38}$$

$$s_{jtp} = s_{j(t-1)p} + D_{jtp} - \sum_{v \in \mathcal{V}_r} \Delta_j f_{jvtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{A.39}$$

$$s_{j0p} = S^0_{jp} + D_{j0p} - \sum_{v \in \mathcal{V}_r} \Delta_j f_{jv0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \setminus \{\mathcal{J}^H\}, p \in \mathcal{P}, \tag{A.40}$$

$$s_{jtp} = s_{j(t-1)p} + \sum_{v \in \mathcal{V}_0} f_{jvtp} - \sum_{v \in \mathcal{V}_r} f_{jvtp} + k_{jtp} - l_{jtp}, \quad r \in \mathcal{R}, j \in \mathcal{J}^H \cap \mathcal{J}_r, t \in \mathcal{T} \setminus \{0\}, p \in \mathcal{P}, \tag{A.41}$$

$$s_{j0p} = S^0_{jp} + \sum_{v \in \mathcal{V}_0} f_{jv0p} - \sum_{v \in \mathcal{V}_r} f_{jv0p} + k_{j0p} - l_{j0p}, \quad r \in \mathcal{R}, j \in \mathcal{J}^H \cap \mathcal{J}_r, p \in \mathcal{P}, \tag{A.42}$$

$$s_{jtp} \geq (S^{max}_{jp} + S^{min}_{jp})/2, \quad j \in \mathcal{J}, p \in \mathcal{P}, t = |\mathcal{T}| - 1, \tag{A.43}$$

$$\sum_{v \in \mathcal{V}} z_{jvt} \leq B_j, \quad j \in \mathcal{J}, t \in \mathcal{T}, \tag{A.44}$$

$$F_j^{min} z_{jvt} \leq \sum_{p \in \mathcal{P}} f_{jvtp} \leq F_j^{max} z_{jvt}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.45}$$

$$S_{jp}^{min} \leq s_{jtp} \leq S_{jp}^{max}, \quad j \in \mathcal{J}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.46}$$

$$0 \leq k_{jtp}, l_{jtp}, \quad j \in \mathcal{J}^{\mathcal{H}}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.47}$$

$$0 \leq s_{jtp} \quad j \in \mathcal{J}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.48}$$

$$0 \leq u_{ijvtp}^{X}, \quad i, j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.49}$$

$$0 \leq f_{jvtp}, u_{jvtp}^{A}, u_{jvtp}^{B}, u_{jvtp}^{W}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, p \in \mathcal{P}, \tag{A.50}$$

$$x_{ijvt} \in \{0, 1\}, \quad i, j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, t + T_{ij} \leq |T| - 1 \tag{A.51}$$

$$z_{jvt}, w_{jvt}, z_{jvt}^{A}, z_{jvt}^{B} \in \{0, 1\}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} \tag{A.52}$$

Tightening of flow variables

$$F_j^{min} z_{jvt}^{B} \leq \sum_{p \in \mathcal{P}} u_{jvtp}^{B}, \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.53}$$

$$\sum_{p\in\mathcal{P}} u^B_{jvtp} \leq z^B_{jvt}(Q_v - F^{min}_j), \quad j \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.54}$$

$$z_{jvt}F^{min}_j \leq \sum_{p\in\mathcal{P}}(u^A_{jvtp} + u^B_{jvtp}), \quad j \in \mathcal{J}^{\mathcal{C}}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.55}$$

$$\sum_{p\in\mathcal{P}} u^W_{jvtp} \leq w_{jvt}(Q_v - F^{min}_j), \quad j \in \mathcal{J}^{\mathcal{P}}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.56}$$

$$w_{jvt}F^{min}_j \leq \sum_{p\in\mathcal{P}} u^W_{jvtp}, \quad j \in \mathcal{J}^{\mathcal{C}}, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.57}$$

$$x_{ijvt}F^{min}_j \leq \sum_{p\in\mathcal{P}} u^X_{ijvtp} \leq x_{ijvt}(Q_v - F^{min}_i), \quad (i,j) \in \mathcal{J}^C, v \in \mathcal{V}, t \in \mathcal{T}, \tag{A.58}$$

Valid Inequalities

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} D_{jtp} + \sum_{p\in\mathcal{P}}(S^{min}_{jp} - S^{max}_{jp})}{\bar{Q}_0} \leq \sum_{i\in\mathcal{J}\cup D_v}\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} x_{jivt}, \quad j \in \mathcal{J}^P, \tag{A.59}$$

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} D_{jtp} + \sum_{p\in\mathcal{P}}(S^{min}_{jp} - S^{max}_{jp})}{F^{max}_j} \leq \sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} z_{jvt}, \quad j \in \mathcal{J}^P, \tag{A.60}$$

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} -D_{jtp} + \sum_{p\in\mathcal{P}}(S^{min}_{jp} - S^{max}_{jp})}{\bar{Q}_r} \leq \sum_{i\in\mathcal{J}}\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} x_{ijv(t-T_{jiv})}, \quad r \in \mathcal{R}, j \in \mathcal{J}_r \cap \mathcal{J}^C, \tag{A.61}$$

$$\frac{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} -D_{jtp} + \sum_{p\in\mathcal{P}}(S^{min}_{jp} - S^{max}_{jp})}{F^{max}_j} \leq \sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}} z_{jvt}, \quad j \in \mathcal{J}^C, \tag{A.62}$$

Appendix B

Pseudocodes

B.1 Pseudocode for RFFO Heuristic

Algorithm 5 R&F & F&O Heuristic

```
function RFFO(m, currentObj)
    for var in m do
        if var is binary then
            var ← continuous
        end if
    end for
    prevObj ← ∞
    while currentObj < prevObj or elapsedTime < timeLimit do
        for c in clusters do
            for var in c do
                if var is originally binary then
                    var ← binary
                end if
                if var is fixed then
                    unfix var
                end if
            end for
            optimize m until feasible solution or timeLimit_1
            if feasible, optimize m until MIPGap or timeLimit_2
            for var in c do
                if var is binary then
                    fix var to solution
                end if
            end for
        end for
        prevObj ← currentObj
        currentObj ← m.objective
    end while
    return solution of m
end function
```

Appendix C

Test Instances

C.1 All Test Instances

Table C.1: All test instances.

Group	Instance no.	Name	$\|\mathcal{T}\|$	$\|\mathcal{R}\|$	$\|\mathcal{J}\|$	$\|\mathcal{VP}\|$	$\|\mathcal{VC}\|$	$\|\mathcal{P}\|$
Time	T1	20T3R3J3V2P	20	3	3	3	3	2
	T2/B1	40T3R3J3V2P	40	3	3	3	3	2
	T3	60T3R3J3V2P	60	3	3	3	3	2
	T4	80T3R3J3V2P	80	3	3	3	3	2
	T5	100T3R3J3V2P	100	3	3	3	3	2
	T6	120T3R3J3V2P	120	3	3	3	3	2
	T7	140T3R3J3V2P	140	3	3	5	3	2
	T8	160T3R3J3V2P	160	3	3	5	3	2
	T9	180T3R3J3V2P	180	3	3	5	3	2
	T10	200T3R3J3V2P	200	3	3	5	3	2
Region	R2	40T2R3J3V2P	40	2	3	3	3	2
	R3/B1	40T3R3J3V2P	40	3	3	3	3	2
	R4	40T4R3J3V2P	40	4	3	3	3	2
	R5	40T5R3J3V2P	40	5	3	3	3	2
	R6	40T6R3J3V2P	40	6	3	3	3	2
	R7	40T7R3J3V2P	40	7	3	3	3	2
	R8	40T8R3J3V2P	40	8	3	3	3	2
	R9	40T9R3J3V2P	40	9	3	3	3	2
	R10	40T10R3J3VP3VC2P	40	10	3	3	3	2
	R11	40T11R3J5VP3VC2P	40	11	3	5	3	2
	R12	40T12R3J6VP3VC2P	40	12	3	6	3	2
	R15	40T15R3J7VP3VC2P	40	15	3	7	3	2
	R20	40T20R3J10VP3VC2P	40	20	3	10	3	2
	R25	40T25R3J12VP3VC2P	40	25	3	12	3	2

Continued on next page

Table C.1 – continued from previous page

Group	Instance no.	Name	$\|\mathcal{T}\|$	$\|\mathcal{R}\|$	$\|\mathcal{J}\|$	$\|\mathcal{VP}\|$	$\|\mathcal{VC}\|$	$\|\mathcal{P}\|$
	R30	40T30R3J15VP3VC2P	40	30	3	15	3	2
Ports	J1	40T3R1J3V2P	40	3	1	3	3	2
	J2	40T3R2J3V2P	40	3	2	3	3	2
	J3/B1	40T3R3J3V2P	40	3	3	3	3	2
	J4	40T3R4J3V2P	40	3	4	3	3	2
	J5	40T3R5J3V2P	40	3	5	3	3	2
	J6	40T3R6J3V2P	40	3	6	3	3	2
	J7	40T3R7J3V2P	40	3	7	3	3	2
	J8	40T3R8J3V2P	40	3	8	3	3	2
Vessel	V1	40T3R3J1V2P	40	3	3	1	1	2
	V2	40T3R3J2V2P	40	3	3	2	2	2
	V3/B1	40T3R3J3V2P	40	3	3	3	3	2
	V4	40T3R3J4V2P	40	3	3	4	4	2
	V5	40T3R3J5V2P	40	3	3	5	5	2
	V6	40T3R3J6V2P	40	3	3	6	6	2
	V7	40T3R3J7V2P	40	3	3	7	7	2
	V8	40T3R3J8V2P	40	3	3	8	8	2
Products	P1	40T3R3J3V1P	40	3	3	3	3	1
	P2/B1	40T3R3J3V2P	40	3	3	3	3	2
	P3	40T3R3J3V3P	40	3	3	3	3	3
	P4	40T3R3J3V4P	40	3	3	3	3	4
	P5	40T3R3J3V5P	40	3	3	3	3	5
	P6	40T3R3J3V6P	40	3	3	3	3	6
	P7	40T3R3J3V7P	40	3	3	3	3	7
	P8	40T3R3J3V8P	40	3	3	3	3	8
Var. Demand	VD1	20T3R3J3V2PVD	20	3	3	3	3	2
	VD2	40T3R3J3V2PVD	40	3	3	3	3	2
	VD3	60T3R3J3V2PVD	60	3	3	3	3	2
	VD4	80T3R3J3V2PVD	80	3	3	3	3	2
	VD5	100T3R3J3V2PVD	100	3	3	3	3	2
	VD6	120T3R3J3V2PVD	120	3	3	3	3	2
	VD7	140T3R3J3V2PVD	140	3	3	5	3	2
	VD8	160T3R3J3V2PVD	160	3	3	5	3	2
	VD9	180T3R3J3V2PVD	180	3	3	5	3	2
	VD10	200T3R3J3V2PVD	200	3	3	5	3	2
	C1.1	40T4R6J8VP3VC3P	40	4	6	8	3	3
	C1.2	60T4R6J8VP3VC3P	60	4	6	8	3	3

Continued on next page

Table C.1 – continued from previous page

Group	**Instance no.**	**Name**	$\|\mathcal{T}\|$	$\|\mathcal{R}\|$	$\|\mathcal{J}\|$	$\|\mathcal{VP}\|$	$\|\mathcal{VC}\|$	$\|\mathcal{P}\|$
	C1.3	80T4R6J8VP3VC3P	80	4	6	8	3	3
	C1.4	100T4R6J8VP3VC3P	100	4	6	8	3	3
	C1.5	40T4R7J8VP3VC3P	40	4	7	8	3	3
	C1.6	60T4R7J8VP3VC3P	60	4	7	8	3	3
	C1.7	80T4R7J8VP3VC3P	80	4	7	8	3	3
	C1.8	40T4R8J8VP4VC3P	40	4	8	8	4	3
	C1.9	60T4R8J8VP4VC3P	60	4	8	8	4	3
Complex	C1.10	80T4R8J8VP4VC3P	80	4	8	8	4	3
	C2.1	50T5R6J3VP3VC2P	50	5	6	3	3	2
	C2.2	100T5R6J8VP3VC2P	100	5	6	8	3	2
	C2.3	100T5R6J8VP4VC4P	100	5	6	8	4	4
	C2.4	50T5R7J4VP4VC2P	50	5	7	4	4	2
	C2.5	100T5R7J8VP4VC2P	100	5	7	8	4	2
	C2.6	50T5R8J4VP4VC2P	50	5	8	4	4	2
	C2.7	100T5R8J10VP4VC2P	100	5	8	10	4	2
	C2.8	50T5R9J5VP5VC2P	50	5	9	5	5	2
	C2.9	100T5R9J10VP5VC2P	100	5	9	10	5	2
	C2.10	40T5R12J6VP6VC2P	40	5	12	6	6	2

C.2 Initial Test Instances

Table C.2: Initial test instances.

Group	Instance no.	Name	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{J}\rvert$	$\mathcal{VP}$	$\lvert\mathcal{VC}\rvert$	$\lvert\mathcal{P}\rvert$
Time	T1	20T3R3J3V2P	20	3	3	3	3	2
	T3	60T3R3J3V2P	60	3	3	3	3	2
	T6	120T3R3J3V2P	120	3	3	3	3	2
Region	R1	40T2R3J3V2P	40	2	3	3	3	2
	R3	40T4R3J3V2P	40	4	3	3	3	2
	R5	40T6R3J3V2P	40	6	3	3	3	2
Ports	J1	40T3R1J3V2P	40	3	1	3	3	2
	J3	40T3R3J3V2P	40	3	3	3	3	2
	J5	40T3R5J3V2P	40	3	5	3	3	2
Vessel	V1	40T3R3J1V2P	40	3	3	1	1	2
	V2	40T3R3J2V2P	40	3	3	2	2	2
	V4	40T3R3J4V2P	40	3	3	4	4	2
Products	P1	40T3R3J3V1P	40	3	3	3	3	1
	P3	40T3R3J3V3P	40	3	3	3	3	3
	P4	40T3R3J3V4P	40	3	3	3	3	4
	P6	40T3R3J3V6P	40	3	3	3	3	6
Var. Demand	VD1	40T3R3J3V2PVD	40	3	3	3	3	2
	VD3	60T3R3J3V2PVD	60	3	3	3	3	2
	VD4	80T3R3J3V2PVD	80	3	3	3	3	2
	VD5	100T3R3J3V2PVD	100	3	3	3	3	2

Complex	C2.1	50T5R6J3VP3VC2P	50	5	6	3	3	2
	C2.2	100T5R6J8VP3VC2P	100	5	6	8	3	2
	C2.3	100T5R6J8VP4VC4P	100	5	6	8	4	4
	C2.4	50T5R7J4VP4VC2P	50	5	7	4	4	2
	C2.5	100T5R7J8VP4VC2P	100	5	7	8	4	2
	C2.6	50T5R8J4VP4VC2P	50	5	8	4	4	2
	C2.7	100T5R8J10VP4VC2P	100	5	8	10	4	2
	C2.8	50T5R9J5VP5VC2P	50	5	9	5	5	2
	C2.9	100T5R9J10VP5VC2P	100	5	9	10	5	2
	C2.10	40T5R12J6VP6VC2P	40	5	12	6	6	2

Appendix D

Results

D.1 Full Results from FCNF vs. AD-RFFO

Table D.1: Full Test Results.

	FCNF			Heuristic		
Instance	Obj. value	Dual value	Dual gap	Obj. value	Solution time [s]	Dual gap
T1	1923443	1870930	2,73%	1947443	540	3,93%
T2	3558781	3249938	8,68%	3578463	5400	9,18%
T3	7359731	6634825	9,85%	7374730	1260	10,03%
T4	8205442	7607152	7,29%	8370848	1020	9,12%
T5	13448282	11781205	12,40%	12855867	660	8,36%
T6	-	-	-	16224597	728	-
T7	-	-	-	14486086	3680	-
T8	-	14540669	-	15779474	1821	7,85%
T9	-	-	-	19436458	4518	-
T10	-	-	-	24014950	1060	-
R2	1576211	1512426	4,05%	1696213	902	10,84%
R3	3558781	3249938	8,68%	3578463	5400	9,18%
R4	7112245	6840150	3,83%	7316245	637	6,51%
R5	8071690	7514840	6,90%	8175462	1219	8,08%
R6	11448502	10815433	5,53%	11760251	3454	8,03%
R7	14456835	13529372	6,42%	14685299	6000	7,87%
R8	17485034	16215261	7,26%	17754992	5640	8,67%
R9	19778141	18601873	5,95%	19961615	1782	6,81%
R10	23286945	22099835	5,10%	23695349	1929	6,73%
R11	-	21909398	-	23666559	3733	7,42%
R12	-	24975878	-	27148192	6242	8,00%
R15	-	32221957	-	35140346	2780	8,30%

Continued on next page

Table D.1 – continued from previous page

	FCNF			Heuristic		
Instance	Obj. value	Dual value	Dual gap	Obj. value	Solution time [s]	Dual gap
R20	-	-	-*	47102258	6832	-
R25	-	-	-*	59590109	6075	-
R30	-	-	-*	70012116	7623	-
J1	3098286	3002102	3,10%	3114947	7	3,62%
J2	3895960	3797399	2,53%	3925021	607	3,25%
J3	3558781	3249938	8,68%	3578463	5400	9,18%
J4	4044916	3692343	8,72%	4020916	5400	8,17%
J5	3639333	3097484	14,89%	3549661	3048	12,74%
J6	4078473	3199041	21,56%	3965400	5793	19,33%
J7	-	3514777	-	4605438	5825	23,68%
J8	-	3493312	-	4509376	6069	22,53%
V1	2091909	2033678	2,78%	2163910	4920	6,02%
V2	2853204	2586601	9,34%	2878917	3000	10,15%
V3	3558781	3249938	8,68%	3578463	5400	9,18%
V4	5239148	5095693	2,74%	5436334	1200	6,27%
V5	5039193	4870288	3,35%	5247425	973	7,19%
V6	7451107	7242851	2,79%	7666283	37	5,52%
V7	7615802	7489478	1,66%	8047802	54	6,94%
V8	10863984	10650258	1,97%	11019985	52	3,36%
P1	3839417	3648156	0.0498	3925016	409	0.0705
P2	3558781	3249938	8,68%	3578463	5400	9,18%
P3	3643065	3458881	5,06%	3719439	3913	7,01%
P4	3549544	3103659	12,56%	3525544	4821	11,97%
P5	3633334	3377941	7,03%	3725657	1235	9,33%
P6	3500860	2883388	17,64%	3308370	3019	12,85%
P7	4154271	3752474	9,67%	4077240	1456	7,97%
P8	4392065	3991163	9,13%	4332064	3025	7,87%
VD1	1688851	1673495	0,91%	1783596	3	6,17%
VD2	3568451	3385570	5,12%	3592451	3608	5,76%
VD3	6128832	5826623	4,93%	6313483	928	7,71%
VD4	11531771	9941955	13,79%	11035629	2035	9,91%
VD5	12673685	10905855	13,95%	12124129	1600	10,05%
VD6	-	14524204	-	15751053	711	7,79%
VD7	-	16904308	-	18018469	1500	6,18%
VD8	-	-	-	19844395	4054	-

Continued on next page

Table D.1 – continued from previous page

	FCNF			Heuristic		
Instance	Obj. value	Dual value	Dual gap	Obj. value	Solution time [s]	Dual gap
VD9	-	-	-	23982931	1497	-
VD10	-	-	-	27371399	3933	-
C1	-	9903310		12922461	1039	23,36%
C2		-	-	29439045	6360	-
C3	-	-	-	40340571	7831	-
C4	-	13879577		17376315	6872	20,12%
C5	-	-	-	37834547	4190	-
C6	-	-	-	17219138	7094	-
C7	-	-	-	38471856	5220	-
C8	-	-	-	22662396	6254	-
C9	-	-	-	47901489	5672	-
C10	-	-	-	20606556	3725	-
C11	-	-	-	5206497	5856	-
C12	-	6973415		8683526	6437	19,69%
C13	-	-	-	15657083	5867	-
C14	-	-	-	21627090	7110	-
C15	-	-	-	6187037	7039	-
C16	-	-	-	9787607	7948	-
C17	-	-	-	17126989	6233	-
C18	-	-	-	7196031	7867	-
C19	-	-	-	15513963	6000	-
C20	-	-	-	20505604	8073	-

*The nodes ran out of memory to create model and start computation

D.2 Variables Generated by FCNF & AD-RFFO

Table D.2: Number of continuous and integer variables.

	FCNF		Heuristic	
Instance	Continuous var.	Integer var.	Continuous var.	Integer var.
T20	6768	2820	3488	1088
T40/B1	16632	6966	8250	2557
T60	27208	11333	13523	4071
T80	37762	15700	18909	5879
T100	48446	20110	24068	7318
T120	58694	24371	29027	8705
T140	78848	32619	41348	12592
T160	91222	37721	48009	14605
T180	102664	42504	54318	16505
T200	115138	47591	60662	18575
R2	8844	3713	4663	1543
R3/B1	16632	6966	8250	2557
R4	24602	10259	12334	3834
R5	33718	14056	16558	5258
R6	40684	16944	20424	6204
R7	50210	20914	25645	7774
R8	57686	24020	28607	8693
R9	65030	27071	31001	10061
R10	72488	30618	34276	11000
R11	87792	36521	42277	13863
R12	98858	41286	51906	15916
R15	128252	53549	68533	20965
R20	-*	-	107326	33989
R25	-*	-	145612	45070
R30	-*	-	185213	59101
J1	8508	3417	6137	1910
J2	12078	4952	8463	2655
J3/B1	16632	6966	8250	2557
J4	22606	9616	11315	3612
J5	29296	12648	14545	4755
J6	36578	15966	16719	1966

Continued on next page

Table D.2 – continued from previous page

	FCNF		Heuristic	
Instance	Continuous var.	Integer var.	Continuous var.	Integer var.
J7	44492	19651	18535	2628
J8	52624	23434	21778	2673
V1	5968	2251	3854	1193
V2	11182	4572	7745	2369
V3/B1	16632	6966	8250	2557
V4	22720	9574	12404	3985
V5	28590	12094	16240	5299
V6	32748	13953	18990	6303
V7	38624	16524	23540	7773
V8	42620	18339	25224	8519
P1	8393	7023	4070	2618
P2/B1	16632	6966	8250	2557
P3	25779	7169	12966	2656
P4	34108	7122	17421	2688
P5	43285	7191	22110	2705
P6	51012	7097	25913	2663
P7	58366	6951	29454	2622
P8	66664	6966	33647	2615
VD1	7060	2953	3747	1168
VD2	17084	7135	8461	2609
VD3	27650	11512	13796	4159
VD4	37498	15609	18332	5585
VD5	48014	19947	23789	7160
VD6	58640	24352	28833	8765
VD7	78608	32569	41160	12554
VD8	90532	37469	47942	14814
VD9	102516	42466	53804	16404
VD10	114792	47478	60172	18324
C2.1	92732	40493	41519	2207
C2.2	224830	97869	108674	16949
C2.3	568072	124648	212195	16772
C2.4	150508	66930	49377	6783
C2.5	342128	151504	124061	20017
C2.6	180110	80801	56916	6740
C2.7	418552	187182	146848	22620

Continued on next page

Table D.2 – continued from previous page

	FCNF		Heuristic	
Instance	Continuous var.	Integer var.	Continuous var.	Integer var.
C2.8	264540	120094	87145	7283
C2.9	592750	268722	205652	22221
C2.10	379454	175599	90910	6247
C1.1	88512	25560	41906	4232
C1.2	145620	42088	66676	7027
C1.3	199452	57718	92420	9886
C1.4	256737	74287	119771	12896
C1.5	106332	31083	47040	5279
C1.6	174513	51084	75509	8637
C1.7	239430	70122	102995	12094
C1.8	163644	48822	54048	5283
C1.9	259635	77523	84562	8457
C1.10	362889	108139	119514	12182

*The nodes ran out of memory to create model and start computation

Chaudhary, Drangsland, Fossum

A Decomposition-Based Matheuristic for the Two-Echelon Multi-Product Maritime Inventory Routing Problem

NTNU
Norwegian University of
Science and Technology