UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MARCELO WUTTIG FRISKE

Solution Methods for a Maritime Inventory Routing Problem

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Profa. Dr. Luciana Salete Buriol
Coadvisor: Prof. Dr. Eduardo Camponogara

Porto Alegre
January 2021

CIP — CATALOGING-IN-PUBLICATION

Friske, Marcelo Wuttig

Solution Methods for a Maritime Inventory Routing Problem / Marcelo Wuttig Friske. – Porto Alegre: PPGC da UFRGS, 2021.

132 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2021. Advisor: Luciana Salete Buriol; Coadvisor: Eduardo Camponogara.

1. Maritime Inventory Routing Problem. 2. Matheuristics. 3. Metaheuristics. 4. Formulations. I. Buriol, Luciana Salete. II. Camponogara, Eduardo. III. Título.

To my parents Eldevir and Rosane, my wife Mariana, and my son Antônio

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

F&O	Fix-and-optimize algorithm
FCNF	Fixed-charge network flow
IRP	Inventory routing problem
LNG	Liquefied natural gas
LNS	Large neighborhood search
MILP	Mixed integer linear program
MIP	Mixed integer program
MIRP	Maritime inventory routing problem
R&F	Relax-and-fix algorithm
RHH	Rolling horizon heuristic
TS	Time-space network

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This thesis presents a matheuristic framework and a metaheuristic approach for solving a Maritime Inventory Routing Problem (MIRP). The problem combines two main components: ship routing and inventory management at ports. Each port has a storage capacity and variable production or consumption rates along with the planning horizon. The vessels differ with respect to capacity, speed, and travel costs. The problem consists of defining a route and schedule for each vessel, besides the amount of product loaded or unloaded in each port visit, while keeping the ports' inventory between lower and upper limits. Constraints on ports inventory and vessel capacity are accounted for the problem, besides side constraints based on a real world scenario. The objective is to maximize the revenue of delivering the product at discharging ports, deducted traveling, and operational costs.

The matheuristic framework is composed of a relax-and-fix algorithm and a fix-and-optimize algorithm. The relax-and-fix algorithm builds an initial solution and consists of dividing the original problem into subproblems solved iteratively. The fix-and-optimize algorithm is responsible for improving the solution, solving partially fixed subproblems derived from a starting solution. The matheuristic framework was tested in two discrete-time formulations, and several formulations components such as additional constraints, preprocessing phase, restriction strategies, and valid inequalities were proposed.

The metaheuristic approach is composed of a multi-start algorithm and a large neighborhood search, being the first proposed method for the MIRP variant considered in this work independent of a mathematical solver.

Tests were carried out on instances from the literature and on modified instances. We evaluated the contribution of the different formulation components of the matheuristc framework, besides different parameter values of the metaheuristic approach, considering the solution quality and processing time. We considered tests with a priori parameter setting and also using an automatic configuration tool. The computational results demonstrated that the proposed methods are potentially effective for solving the MIRP when applied to a public dataset, obtaining new best-known solutions, and providing solutions for instances in which no attempts to solve them were presented in the literature.

Keywords: Maritime Inventory Routing Problem. Matheuristics. Metaheuristics. Formulations.

Métodos de Solução para um Problema de Roteamento de Inventário Marítimo

RESUMO

Esta tese de doutorado apresenta matheuristicas e metaheuristicas para resolver um Problema de Roteamento de Inventário Marítimo (*MIRP - Maritime Inventory Routing Problem*) de produto único. O problema combina dois componentes chave: o roteamento de navios e a gestão de estoque nos portos. Cada porto possui uma capacidade de estocagem e produz ou consome determinada quantidade de produto ao longo do horizonte de planejamento. A frota de navios é heterogênea, sendo que os navios diferem entre si por capacidade, velocidade, e custos de navegação. O problema consiste em definir uma rota e um escalonamento para cada navio, que é composto por uma sequência de visitas a portos de carregamento e descarregamento em períodos de tempo específicos. Além disso, é necessário definir em cada visita a quantidade a ser carregada/descarregada pelo navio. São consideradas restrições de capacidade de estoque nos portos e capacidade dos navios, além de restrições auxiliares baseadas em cenários do mundo real. O objetivo é maximizar a receita através da entrega de produto nos portos de descarregamento, subtraindo os custos operacionais e de viagem dos navios. A estrutura matheurística é composta por um algoritmo *relax-and-fix* e por um algoritmo *fix-and-optimize*. O primeiro constrói uma solução inicial e consiste em dividir o problema em subproblemas que são resolvidos de forma iterativa. O segundo é responsável por melhorar a solução obtida pelo primeiro, resolvendo problemas inteiros mistos parcialmente fixados de forma iterativa. A estrutura matheuristica foi testada em duas formulações de tempo discreto: um modelo de rede espaço-tempo e um modelo de fluxo de carga fixa. Além disso, diversos componentes para foram propostos, tais como restrições adicionais, desigualdades válidas e pré processamento.

A solução metaheuristica é composta de um algoritmo *multi-start* e algoritmo *large neighborhood search*, sendo o primeiro método a ser proposto para a variante do MIRP considerada neste trabalho que não depende de um resolvedor matemático para obter soluções. Os testes computacionais foram executados sob instâncias da literatura e instâncias modificadas. Nós avaliamos a contribuição de diferentes componentes das formulações da estrutura matheuristica, além de diferentes valores de parâmetros da abordagem metaheuristica considerando a qualidade da solução obtida e o tempo de execução. Foram considerados testes com a definição de parâmetros a priori e também utilizando uma ferramenta de configuração automática de parâmetros. Os resultados demonstraram que os métodos propostos são potencialmente efetivos para resolver o problema quando aplicados a um conjunto de instâncias públicas, obtendo novas melhores soluções conhecidas, e fornecendo soluções para instâncias nas quais não foram apresentadas na literatura tentativas de solução.

Palavras-chave: Problema de Roteamento de Inventário Marítimo, Matheuristica, Metaheuristica, Formulações.

1 INTRODUCTION

Maritime transportation is the most significant transportation mode concerning large volumes. It is responsible for over 80% of the total volume of world trade. According to UNCTAD (2019), the amount transported by this mode in 2018 was approximately 11 billion tons, a figure that has been growing over several years, although it lost momentum in the cited year. The world fleet capacity increased by 2.6% from the beginning of 2018 until the beginning of 2019. Considering the Brazilian scenario, the cargo movement, including bulk, containers, and general cargo, had an increase of 8.3% from 2016 to 2017 (ANTAQ, 2017).

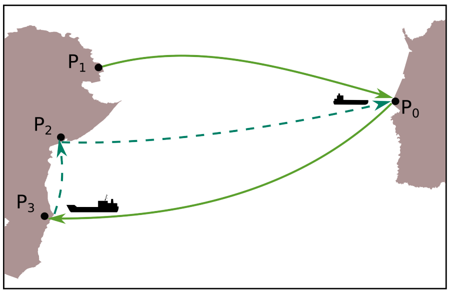
Maritime transportation's growing perspectives lead to more challenging problems concerning fleets, cargoes, and inventory management. Thus, the use of optimization strategies is critical in this sector since maritime logistics is a capital-intensive industry, where a relatively small improvement in its operations can provide a significant economy (AGRA et al., 2013). The use of optimization tools can improve maritime transportation in different aspects, such as cost reductions, profits, fleet utilization, and lower emissions levels, among others.

This work studies the Maritime Inventory Routing Problem (MIRP), a challenging problem that deals with routing and scheduling vessels and inventory management at ports. It is a relatively recent problem that has gained attention in the last decade. The MIRP is an extension of the Inventory Routing Problem (IRP), a road-based problem that combines the well-known Vehicle Routing Problem (VRP) with inventory management. Although there is no exact formal definition of the MIRP as diverse variations can be found in the literature, a basic definition can be stated as follows: Given a finite planning horizon, a fleet of vessels with different characteristics such as cargo capacity and speed, and given a set of loading and discharging ports, each one with storage capacity and production and discharging rates, one must decide for each vessel: **i) the routing**: a sequence of ports visited; **ii) the scheduling**: times when it will travel between ports on the route, and the time at which loading and unloading operations will take place; and **iii) the amount** to be loaded and discharged in each port visit. Constraints on vessels' capacity and ports' inventory should be respected. The problem objective usually aims to minimize transportation and operation costs.

Figure 1.1 illustrates an example of a solution for a MIRP with one loading port and three discharging ports and two vessels for transporting one product type. The left side represents each vessel's route, and the right side is the corresponding schedule, informing the time at which each port is visited and the amount loaded or discharged in that port.

Figure 1.1: Example of vessels route and schedule for a basic MIRP.

Route



Schedule

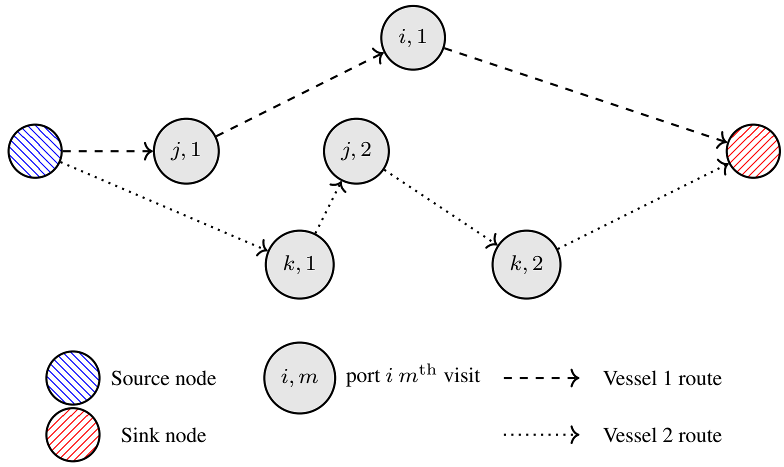
Vessel 0	**Port**	P_1	P_0	P_3
	Time	1	10	20
	Discharge	150	-	150
	Load	-	150	-
Vessel 1	**Port**	P_3	P_2	P_0
	Time	1	5	18
	Discharge	75	25	-
	Load	-	-	100

Source: From the author (2020).

Maritime transportation problems, such as the MIRP, involve different operations modes and planning levels. According to Lawrence (1972), there are three operations modes in maritime transportation: liner, tramp, and industrial shipping. In liner shipping, vessels operate according to a published route and schedule, similar to a bus line. In this case, the demand for vessel services depends on their schedules. Containers and general cargo vessels belong to this mode. Tramp shipping is similar to a taxi cab, where vessel owners follow available cargoes, engaging contracts of affreightment, which determines periodic travels for pick up and deliver cargo between two or more ports, paying a defined amount per ton carried. Tankers, dry bulks, and refrigerated vessels usually are operated in this mode. In industrial shipping, the operator owns the cargo and controls the fleet that can be own and chartered. This operation can be found in vertically integrated companies, such as oil and gas, chemicals, and ores. While liner and tramp shipping's objective usually aims to maximize the profit obtained by transporting the cargoes, industrial shipping aims to minimize transportation costs.

Concerning the planning level, maritime transportation can be divided into strategic, tactical, and operational levels (PAPAGEORGIOU et al., 2014b). Strategic planning involves decisions that will be taken for a horizon of one to twenty years. Strategic problems can be related to renewing a fleet of vessels based on actual and future operations. The tactical planning involves a time horizon of months, up to one year, where decisions involve vessel routing and product distribution. The operational level concerns planning of some weeks up to a few months, including problems with a minor time granularity, such as allocating vessels to berths in a port.

Figure 1.2: Continuous-time model illustration for a network with three ports and two vessels



Source: From the author (2020).

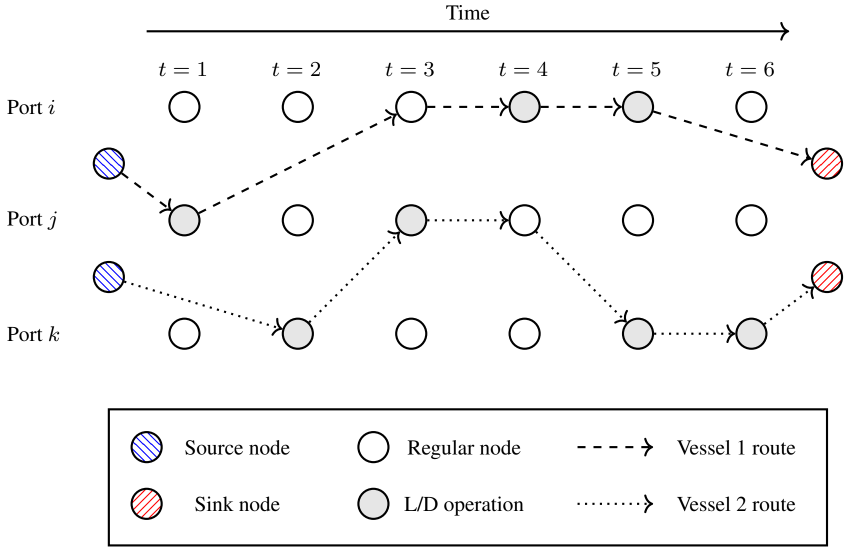
MIRP can be considered an industrial shipping problem with a strategic/tactical level, where vertically integrated companies hold the cargo and the fleet. Also, this problem can be applied either in short-sea or deep sea-configurations. In short-sea, the time spent by the vessels operating at ports is greater than the time sailing between them, characterizing a regional problem. In contrast, in deep-sea, the traveling times are much higher than operating times, describing intercontinental voyages.

1.1 Discrete and Continuous-time Formulations

The MIRP can be modeled considering continuous or discrete-time formulations. In continuous-time models, the events (travels, loading, and discharging operations) can occur at any moment of the time horizon. In discrete-time, the planning horizon is divided into time periods with equal intervals, and events can only occur at such time periods. Figures 1.2 and 1.3 illustrate a continuous and a discrete time models, respectively. They show a possible solution for a MIRP with three ports (i, j, k) and two vessels.

In Figure 1.2, each vessel departs from the dummy source node, and vessel 1 performs the first visit to port j, while vessel 2 performs the first visit to port k. A port visit can consider waiting time, preparation time, operation (loading/discharging) time, among others. Vessel 1 then departs to port i, performing the first visit at this port, and vessel 2 departs to port j, which received its second visit. Finally, vessel 1 finishes its route arriving at the dummy sink node, while vessel 2 returns to port k in the second visit,

Figure 1.3: Discrete-time model illustration for a network with three ports and two vessels



Source: From the author (2020).

and after, complete its route.

According to Figure 1.3, vessel 1 starts its route arriving at port j in time period $t = 1$, performing an operation. Then, it takes two time periods to travel to port i, arriving at time $t = 3$. After, vessel 1 waits one period before operating in two consecutive time periods and ending its route reaching the sink node. Vessel 2 arrives at port k at time period 2, operates, and departs in the direction to port j in the same time period. When reaching port j, it operates immediately, and after waits one time period before departing again to port k. After operating two consecutive times at port k, vessel 2 ends its route at time $t = 6$.

The continuous-time formulations usually have fewer variables and constraints than discrete-time models considering the same problem instance. On the other hand, discrete-time formulations can easily accommodate different aspects, such as the variable production/consumption rates at the ports or multiple berths in each port. It is also possible to use hybrid models, combining both continuous and discrete models.

MIRP formulations can also differ mainly between arc and path flow models. In an arc-flow model, the description of a vessel route is a sequence of variables representing voyages between pairs of ports. While in path-flow models, a unique variable describes the entire route of a vessel. There are also formulations based on patterns or duties (e.g. ANDERSSON; CHRISTIANSEN; DESAULNIERS, 2016), where a variable is associated with a sequence of port visits, and the entire route can be described by the use of one or more duties.

1.2 Difference with Other Problems

MIRPs differs from the classical inventory routing problem in several aspects. As inventory routing problems are road-based, the planning horizon usually consists of hours up to a few days. MIRPs can consider time horizons up to one year, as traveling and operation times are large. On the other hand, the time period's granularity can be more refined in IRP than in MIRPs. In the classical IRP, vehicles start and end their routes at a single central depot, while in MIRPs each vessel can start and end its route from any place, either a port or a point in the sea. Unlike IRP, where the fleet of vehicles are generally homogeneous, MIRPs consider heterogeneous fleets, with different capacities, costs, speeds, and technologies. These differences occur because vessels have a long lifetime (20-30 years). Also, vessels visit only a few ports when loading or discharging since the operational cost and time tend to be higher. In IRPs, customers usually consume small amounts of products, and thus the number of stops of vehicles can be higher before returning to the depot.

A similar problem to MIRP is the cargo routing problem. Unlike MIRPs, where the ports visit and amounts operated by a vessel should be defined, they are given as input data by cargoes in cargo routing problems. Cargo consists of a determined amount of product to be loaded and discharged at specific ports. The inventory management part of the problem establishes time windows for pickup and delivery of each cargo. The problem consists of defining which vessels will carry each cargo and when they will be collected and delivered. Cargo routing problems that not consider split deliveries are more constrained than MIRPs as the number of port visits is known in advance, making MIRPs more challenging than the cargo routing problem (PAPAGEORGIOU et al., 2014b).

1.3 Literature Reviews

Literature surveys on maritime transportation that covers maritime inventory routing problems are available. Reviews on maritime transportation were presented about every decade by Ronen (1983), Ronen (1993), Christiansen, Fagerholt and Ronen (2004), Christiansen et al. (2013). They state that the number of works on maritime transportation optimization has doubled in each decade, following the sector, which is experiencing a continuous increase. A dedicated book chapter of Christiansen et al. (2007) describes a broad investigation and classification among the maritime problems. A revision of routing and inventory management problems considering road and maritime transportation is presented in Andersson et al. (2010) and Coelho, Cordeau and Laporte (2013). The review of Christiansen and Fagerholt (2014) presents different industrial and tramp shipping problems, including a basic formulation and illustrative examples. A review of single-product deterministic MIRPs with a core model is presented by Papageorgiou et al. (2014b). Psaraftis and Kontovas (2013) present an exclusive study on speed optimization for maritime transportation problems. Psaraftis (2019) discusses the increase of studies in theoretical maritime transportation problems focused only on the solution approach than its practical application in a real-world scenario.

1.4 MIRP Variants

Each MIRP variant can consider several characteristics and constraints. These characteristics are combined and form a specific variant. The most common characteristics are:

- *Production and consumption rates at ports*: They can be variable or constant along the planning horizon, usually inducing the use of discrete or continuous-time formulations, respectively.

- *Single or multi products*: In the single-product version, only one product type is considered, and a port is exclusive a production or a consumption port. While in multi-product MIRPs, a port can consume a subset $\mathcal{P}^{\mathrm{C}}$ of products, and a at the same time produces subset of products $\mathcal{P}^{\mathrm{P}}$, such that $\mathcal{P}^{\mathrm{C}} \cap \mathcal{P}^{\mathrm{P}} = \emptyset$.

- *Draft limits*: The vertical distance between the waterline and the bottom of the keel of a vessel is known as the draft. It can vary according to the amount of cargo loaded at the vessel. Thus, this restriction can impose that some vessels can only enter or leave a specific port if the cargo amount is less than a given value.

- *Soft inventory bounds*: Due to the more uncertainty in maritime transportation than other transportation modes, the inventory at ports can be considered soft constrained, paying a penalty amount when the inventory constraints are violated.

- *Number of sequential loading or discharging operations*: In most of the real scenarios, a vessel must pay a port fee to dock in a port for operating. The operation also takes a relatively long time at berth. Thus, vessels usually are restricted to visit only a sequence of few ports of the same type during its route to minimize cost and prevent a possible deviation of the schedule due to unpredictable events.

Uncertainty is also present in MIRPs. The change in weather conditions can influence the navigation time between ports. Simultaneously, variations on the market can affect either production and consumption rates, as well as the sailing and operation costs due to the fuel prices. For dealing with uncertainty, some works consider soft constraints (AGRA; CHRISTIANSEN; DELGADO, 2013; PAPAGEORGIOU et al., 2014b), use stochastic programming strategies, where different scenarios are considered (AGRA et al., 2016; AGRA et al., 2015), or use robust optimization techniques (ZHANG et al., 2015; AGRA et al., 2018).

A particular class of MIRPs known as liquefied natural gas inventory routing problem (LNG-IRP) differs from the standard MIRP due to specific constraints. The problem deals with the supply chain of natural gas, and the loading ports correspond to liquefaction plants, where the gas is cooled to reach the liquid state. Then, vessels with special tanks transport the LNG to the regasification terminals (discharging ports). However, the liquefied gas evaporates at a specific rate during the voyages (boil-off). This boil-off is used as fuel to keep the tanks refrigerated. Thus, the LNG amount loaded into the vessel at a loading port is not the same that the quantity unloaded in the discharging ports (GRØNHAUG; CHRISTIANSEN, 2009). Another difference to the MIRP is that the production and consumption rates can be considered variables instead of input parameters (e.g. GRØNHAUG; CHRISTIANSEN, 2009; ANDERSSON; CHRISTIANSEN; DESAULNIERS, 2016). Additionally, LNG companies usually work with long-term contracts that must be fulfilled, where pre-specified LNG quantities should be delivered during the planning horizon. It is also possible to sell LNG for spot markets for improving the profit (e.g. GOEL et al., 2012).

1.5 Solution Methods

According to Andersson et al. (2010), Song and Furman (2013), most of the MIRPs discussed in the literature are particular variations of the problem, and tailor-made methods are used to solve them, sometimes taking advantage of the problem structure. Thus, different solving approaches can be found in the literature.

Exact methods usually combine formulation improvements such as valid inequalities and constraints tightening. Methods such as branch-and-cut, branch-and-price, and branch-and-cut-and-price algorithms can usually solve to optimality small real-sized instances. Some examples are Christiansen (1999), Andersson (2011), Engineer et al. (2012), Hewitt et al. (2013).

Pure heuristic or metaheuristic methods are less frequently presented since obtaining even a feasible solution is a challenge for MIRPs. Some proposed methods are constructive heuristics combined with genetic algorithm (CHRISTIANSEN et al., 2011), tailor-made heuristics (SISWANTO; ESSAM; SARKER, 2011), and particle swarm optimization (DE et al., 2017).

Matheuristic methods combine exact methods in a heuristic framework. Their use for solving different MIRPs is relatively common, as they demonstrate that high-quality solutions can be found in relatively short processing time. Some matheuristics used for solving different MIRP variations are rolling horizon heuristics (RHH) (RAKKE et al., 2011; AGRA et al., 2014), tailor-made decompositions (PAPAGEORGIOU et al., 2014a; HEMMATI et al., 2016) and relax-and-fix algorithms (UGGEN; FODSTAD; NØRSTEBØ, 2013).

1.6 Literature on MIRP

Although the study on MIRPs and maritime transportation, in general, has grown, it is less established than similar problems such as the classical IRP. A possible reason is that the maritime industry tends to be more conservative. There are few organizations operating vessels compared to those that operate vehicles, which can justify the lower attention in this kind of problem (RONEN, 1983). The lack of available instances for testing algorithms are another reason that can limit the research. Most of the works on MIRP are based on real scenarios, and the data are not publicly available, sometimes because of confidentially terms. Thus, different works that use the same instance sets usually are from the same research group, rare in maritime transportation. To provide a public set of benchmark instances, Papageorgiou et al. (2014b) proposed a library for the problem, named MIRPLIB (PAPAGEORGIOU, 2013), providing instances for three classes of MIRPs, making possible their use for testing different solution approaches.

1.7 Thesis Contribution

This thesis presents and analyzes matheuristic and metaheuristic methods for solving the Maritime Inventory Routing Problem proposed by Papageorgiou et al. (2014a). The matheuristic framework is composed of a relax-and-fix algorithm to build a solution and a fix-and-optimize algorithm with different strategies to improve it. The metaheuristic method consists of a multi-start algorithm and a large neighborhood search.

The main contributions of this work can be listed below:

- We present a survey on the main works on MIRPs, classifying them according to specific characteristics;
- A matheuristic framework was proposed for the presented MIRP variant;
- Different formulations components are proposed, such as additional constraints, preprocessing phase, restriction strategies, and valid inequalities;
- We present an individual analysis of each formulation component and F&O strategy considering processing time and solution quality;
- We proposed a metaheuristic approach for the presented MIRP variant;
- The metaheuristic method can solve long planning horizon instances, in which no attempts to solve them were presented in the literature;
- We tested the algorithms with a priori parameters settings and also with an automatic configuration tool;
- Both solutions methods obtained new best-known solutions for the tested instances.

The work along this thesis originated the following papers:

- **Friske and Buriol (2017)** - Proposed the R&F and F&O for solving the MIRP time-space network formulation (conference paper);
- **Friske and Buriol (2018)** - Proposed the R&F and F&O for solving the MIRP fixed-charge network flow formulation (conference paper);
- **Friske and Buriol (2020)** - Proposed the metaheuristic approach for solving the MIRP (conference paper);

- **Friske, Buriol and Camponogara (2020)** - Proposed new formulation and algorithmic improvements for the matheuristic framework, comparing the performance of both TS and FCNF models (journal paper - submitted for review).

1.8 Thesis Content

The remainder of this thesis is organized as follows. The literature review on the Maritime Inventory Routing Problem and its variations are presented in Chapter 2. Chapter 3 presents the problem description and two mathematical formulations. The proposed solutions methods are presented in Chapter 4. Computational results are reported in Chapter 5. Finally, Chapter 6 presents the conclusions of this thesis, including possible future research on the problem.

2 RELATED WORK

Several works on MIRPs are dedicated to developing models for specific problem variations, usually related to real cases. As a result, there are few comparisons between solution strategies for each variation. This will be demonstrated in this chapter, which discusses the literature on the MIRP, starting from a historical context, presenting some pioneer works on maritime transportation. Next, a review of the works on the Maritime Inventory Routing Problem is presented, being separated into two main characteristics: constant and variable production and consumption rates, and single and multi-products MIRPs. At the end of the chapter, the cited works are summarized in two tables.

2.1 Pioneering works on MIRP

Optimizing maritime transportation is accounted for since the fifties, starting in the postwar. These works were simpler than the actual MIRP variations, as they considered different assumptions, and that most of the information was given in advance. Thus, few decision variables were needed to solve such problems. This section presents a temporal evolution of maritime transportation works starting from the first works in the area until the first work that can be classified as a MIRP.

The work of Dantzig and Fulkerson (1954) is one of the most known pioneers' works accounting maritime logistics. They proposed a linear programming model to define the minimum number of oil tankers to meet a fixed schedule of the Navy. The model assumes that each vessel performs a single loading and unloading operation, being solved as a transportation problem. A similar work proposed by Flood (1954) also considered the objective to minimize the ballast sailing leg, i.e., the total distance of voyages when a vessel is empty. A fixed fleet size was assumed, and the problem is also solved as a transportation problem. The US Navy ordered the works of Dantzig and Fulkerson (1954), Flood (1954). Note that they deal only with the routing of vessels. One decade later, in Briskin (1966), a Tanker Scheduling Problem was presented, where several discharging ports can be visited in sequence, considering a cluster of ports, and allowing vessels to perform partial discharging. The problem is to determine the best delivery dates and the volume to be discharged in each port, such that the quantity delivered at the cluster is equal to the vessel capacity operated in that cluster. The problem also considers the inventory capacity at ports in each period, and the objective is minimizing voyage costs.

Dynamic programming was used to determine the delivery dates and the discharged volumes, and a transportation model determines the schedule of each tanker. In this problem, it is possible to observe two aspects in common to the MIRP: the scheduling of vessels and inventory management.

Appelgren (1969) proposed a Dantzig-Wolfe decomposition for a ship routing problem related to tramp shipping. The problem consists of assigning contracted cargoes to vessels to maximize the revenue. Optionally, spot cargoes can be carried to improve profit. The cargoes have as parameters the loading and discharging ports and predetermined times to be loaded and unloaded. A column generation algorithm was used to solve the problem, while a dynamic programming algorithm solved the subproblems. Although the column generation algorithm solves the linear relaxation of the model decomposition, the algorithm's solution is fractional in less than 1-2% of the cases. Computational experiments were performed considering 40 ships, 50 cargoes, and a horizon of four months. Just two percent of the solutions were fractional, and for dealing with them, a subsequent work proposed a branch-and-bound technique (APPELGREN, 1971). These works originated what is known today as Cargo Routing Problem, a similar problem to the MIRP, in which it is not necessary to decide the amounts to be loaded and unloaded. Bellmore, Bennington and Lubore (1971) extended the work of Appelgren (1969) by considering different types of tankers that can be partially loaded. The problem consists of scheduling and routing the fleet to maximize the total utility, which is related to desirability and delivery cost. Although no results were presented, suggestions of decomposition using branch-and-bound and network subproblems were made. McKay and Hartley (1974) considered a more general tanker scheduling problem with multiple products for the Defense Fuel Supply Center and the Military Sealift Command. The problem is to define a route for each vessel, in which cargoes are assigned to obtain a minimum cost delivery to meet the schedules. Differently from Dantzig and Fulkerson (1954), multiple deliveries and loadings were allowed, and the cargoes are not pre-specified, but demand at consuming ports is given to determine calendar dates. There were different buying costs for the cargoes in the loading ports, and constraints on vessel capacity and weights are imposed. They proposed an arc-flow formulation and a restructured path-flow formulation where routes are decision variables. The solution method consists of building feasible routes based on specific rules, solving the problem in a relaxed form, and applying a special rounding scheme.

A ship routing problem with a planning horizon of up to two weeks was proposed by Ronen (1979). The problem aims to assign cargoes available from a single origin to a heterogeneous vessel fleet, to minimize shipping and demurrage costs. The problem was formulated as a non-linear mixed-integer program. Three different routing algorithms were tested to solve the problem, a heuristic, an exact method, and a biased random routes generation. The last one obtained the best results.

A decision support system for scheduling bulk vessels was described in Stott and Douglas (1981), where the problem was treated as a linear programming model. In such a model, specified voyages with backhauls and vessels chartering are allowed, and the objective aims to minimize the operating costs while meeting loading quantities.

To the best of our knowledge, the work of Miller (1987) was the first one that deals with the routing and scheduling of vessels while keeping control of the inventories at the ports, characterizing it as a MIRP. The proposed MILP flow model minimizes the operating and traveling costs. The problem involves defining each vessel's route between a loading plant and several discharging terminals, ensuring that the inventory of different products in each terminal lies between minimum safety stock and a maximum capacity. Only one vessel can operate in the loading port and the terminals in each time period. Due to the problem size (4 vessels, approximately 30 terminals, 20 products, and a planning horizon of 18 months with a day granularity), a heuristic creates a solution selecting a pre-defined set of candidates routes. For improving the solution, a try-and-error approach was used, swapping among the candidate routes. Also, automatic and manual routines were used to improve the schedules. The automatic routine consists of fixing routing variables and optimizing delivery quantities and vice-versa, while the manual routine possibilities the scheduler to make some changes in the solution, but the system is not able to verify the feasibility of these changes. No computational results were presented. However, the authors conclude that the generated schedules proved to be satisfactory.

Another well-known pioneer work on MIRP is the work of Christiansen (1999), named "Inventory Pickup and Delivery Problem with Time Windows". She combined inventory management with ship routing for the transportation of ammonia between production and consumption ports. The problem considers a heterogeneous fleet of vessels for transportation, while each port has different production/consumption constant rates and storage capacities. Vessels can visit just a subset of ports, and the operation amounts must lie between lower and upper limits in each port visit. The ports have time windows in which they can receive a vessel, and the time to load or unload depends on the amount of product and the port. A dummy ship is considered when a vessel cannot arrive at a port in a specified time window. Also, it is assumed that each port has one berth. The problem objective is to minimize transportation costs, including port fees, canals passing, and fuel consumption. An arc-flow model was presented, followed by a Dantzig-Wolfe decomposition, which is used in a column generation algorithm. A subproblem is accounted for each vessel, being solved as the shortest path problem by a dynamic programming algorithm described in Christiansen and Nygreen (1998b). For obtaining optimal solutions, a branch-and-price algorithm was proposed with branching on flow variables and time windows intervals, which are described in Christiansen and Nygreen (1998a).

2.2 Problem Variants

Although the number of works on MIRP has grown, there is no clear name-classification of such problem variants, such as in the Vehicle Routing Problem, where variants are clearly defined, such as Vehicle Routing with Time Windows, multi-depot vehicle routing problem, among others. This occurs because maritime problems are naturally less structured, as there is more variety in the problem characteristics and operating environments (RONEN, 1993). The first use of the term "Maritime Inventory Routing Problem" was in the work of Al-Khayyal and Hwang (2007). However, the name "Marine Inventory Routing Problem" was already defined by Ronen (2002) for the same problem. Some recent works also use other names for defining the problem because of a specific aspect. For example, in Agra, Christiansen and Delgado (2013), Agra et al. (2014) the problem is called "Short-sea inventory routing problem" (SSIRP) as it deals with the transportation of cargoes in an archipelago. When the transported product is the liquefied natural gas (LNG), the problem usually is called "LNG inventory routing problem" (GRØNHAUG et al., 2010; ANDERSSON; CHRISTIANSEN; DESAULNIERS, 2016). These problems deal with a specific constraint that accounts for the evaporating of the natural gas in the vessel tanks. However, the routing, scheduling, and inventory management are substantially the same as in other MIRPs.

This section discusses the different MIRPs variations that are available in the literature and the solutions methods used. Tables 2.1 and 2.2 summarize the main characteristic of each work.

The works are classified into two disjoint sets: the problems where the production/consumption rates are **constant** along the planning horizon, and the problems where these rates are **variable**. Each set is partitioned into two subsets: the works which consider **single product** and those which consider **multi-product** MIRPs.

2.2.1 Constant Production/Consumption Rates at Ports

This section's works present MIRPs that production and consumption rates at loading and discharging ports are constant along the planning horizon. They can consider long-term problems. The variation in the production and consumption rates along the planning horizon is harder to estimate due to the problem's uncertain nature. Thus, a constant rate can be used based on previous average values.

2.2.1.1 Single-product MIRPs

The work of Jiang and Grossmann (2015) explored continuous and discrete formulations for a basic version of a MIRP. The problem considers sailing costs as a fixed part plus a variable part, which depends on loads of the vessels, and the same occurs for the operating costs and the time to operate. The vessels cannot wait at a port for longer than a time limit. Each vessel has a loading/discharging rate at each port, including a preparation time between the arrival and the start of the port's operation. It considers a simplification where a binary parameter predetermines the route of each vessel. Two continuous models were presented: one based on time slots, which allows only a single berth at ports, and the other based on event points to handle two parallel berths. The computational experiments were performed on randomly generated instances with the CPLEX solver for comparing the formulations. In the instances with one berth in each port, the continuous-time models performed better in CPU time and objective value. When parallel docks instances were considered, the continuous model based on event points obtained significantly better objective functions than the discrete model. However, the CPU time required depends on the particularity of the instances.

A case of study for a vertically integrated company in Norway is presented by Agra et al. (2017) for introducing vendor-managed inventory (VMI) policies. They consider a MIRP for transporting feed from factories to salmon farmings on the Norwegian coast. Although the real case constitutes one feed factory and several products, the problem is assumed to have more than one factory and a unique product. The capacity of the ship is assumed to be greater or equal to the feed factories. Thus, when a vessel visits a factory, it will load all the inventory of the factory. Also, a minimum time between the visits of vessels is considered at the fish farms and feed factories. As the factories have no sufficient capacity to deliver feed to the farms, it is possible to buy feed externally. The continuous-time model is reformulated for improving branch-and-bound efficiency using the special ordered set of variables of type 1 and tightening the bounds by valid inequalities. Additionally, two matheuristics based on practical aspects of the problem were proposed to obtain feasible solutions and improve the current solution.

Zhang et al. (2015) studied a robust MIRP with time windows and stochastic travel and port times. The inventory management was accounted only for loading ports, while at discharging ports, a contracted quantity should be delivered fairly evenly over the entire planning horizon. A simplification assumes that vessels always perform full loads and unloads. A difference from other MIRPs is that the length and placement of the time windows are decision variables. The problem was formulated as a two-stage stochastic, while a two-phase heuristic approach was used to solve it.

Agra et al. (2016) considered a continuous-time model with random traveling times between ports due to the high level of uncertainty of weather conditions. For each visit, a time window is defined such that a vessel can only operate between the start and the end of the time windows. Also, a minimum interval between visits is considered, and a penalization cost is paid for violating the ports' inventory limits. The problem was modeled with a two-stage stochastic programming problem with recourse. The proposed solution method is a MIP-based local search heuristic based on the local branching technique, compared with a decomposition algorithm based on the L-shaped algorithm. A study on the variation of the penalty costs showed that the integrality gaps increase when they are high, turning the decomposition algorithm more unstable.

A general optimization procedure immune to a certain number of delays due to variable sailing times is proposed by Agra et al. (2018). A mandatory number of port visits is given as input, and each visit considers a time window. After the uncertainty is revealed in the model, only the start time of service (load or unload operation) and inventory levels can be adjusted. The solution method consists of decomposing the problem into a master problem, where each robust constraint is considered for a small subset of scenarios. When a feasible solution is found, an adversarial separation problem is solved to verify if a scenario leads to infeasibility. Several improvements in the decomposition strategy are proposed, and an iterated local search based on a local branching scheme is used to obtain solutions for all instances tested.

The work proposed by Rodrigues et al. (2019) compared static and conservative formulation approaches for MIRP. The static model tends to present a lower cost but is not failure tolerant due to weather conditions or congestion in the ports. On the other hand, a conservative plan has failure tolerance but higher costs. The main objective was to compare different techniques to handle uncertainty in the MIRP, such as inventory buffer (in a deterministic model), robust optimization (AGRA et al., 2018), and stochastic optimization with recourse (AGRA et al., 2015). They also proposed two formulations using the concept of conditional value-at-risk to measure the risk of violating inventory bounds.

2.2.1.2 Multi-product MIRPs

Al-Khayyal and Hwang (2007) extended the work of Christiansen (1999) by considering multi-products and dedicated product compartments in the ships. Tests on random instances demonstrated the need for specialized algorithms to efficiently solve the problem, mainly when allowing multiple berth capacities. The problem of different fuel oil product distribution on the Cape Verde archipelago was presented in Agra, Christiansen and Delgado (2013). In this variation, inventory management is taken into account only in the discharging ports, as the inventory at the loading port is assumed to be sufficiently large. The model combines continuous and discrete-time formulations for dealing with multiple time windows at ports. As the problem considers short-sea, more details are accounted for port operating times, which vary according to each product and the quantity loaded/discharged. Several formulation improvements are considered for solving the real instances to optimality, such that tightening constraints, extending formulations, and valid inequalities. Also, a study was conducted on modeling dedicated compartments in vessels and ports for each product.

Differently from Al-Khayyal and Hwang (2007), Li, Karimi and Srinivasan (2010) proposed a continuous-time model in which more than one vessel can operate simultaneously at a port. The problem is related to the transport of chemicals of a multinational company, where ports consume and produce different products. External suppliers meet the demand of products that are not produced by their company ports, but inventory constraints are not considered at these supplier ports. The model was evaluated with four examples, and the solutions and processing times are compared with the works of Al-Khayyal and Hwang (2007), Christiansen (1999).

The work of Siswanto, Essam and Sarker (2011) introduced the use of undedicated compartments into the vessels, meaning that a compartment can carry different products along the planning horizon. Since the products are non-mixable, only one product can be carried in the compartment at a time. Because ships' capacity compartments are larger than the unloaded demands, partial unloading is allowed, and a ship can only be loaded again after becoming empty. Due to the higher computational time for solving the model, a multiple-heuristic approach was used. In this method, the problem is divided into four sub-problems, and different selection rules for each subproblem are used to build the solution. The heuristic solved until optimality 67% of instances that the MILP solver proved optimality. Also, no combination of rules was dominant over all instances.

Agra et al. (2014) proposed a hybrid heuristic for the continuous time arc load flow formulation presented in Agra, Christiansen and Delgado (2017). Improvements on the problem formulation by valid inequalities and tightening constraints were proposed. The solution method consists of a rolling horizon framework, where feasibility pump heuristic was used to find integer solutions, and local branching was used for improving such solutions. The feasibility pump heuristic consists of a rounding scheme of the problem relaxation, while local branching consists in finding a different solution from a previous feasible solution by allowing a limited number of variables to change their value. The computational results showed the method's effectiveness, which is capable of solving instances with six months planning horizon, besides obtaining better solutions than a default rolling horizon heuristic.

The work of Agra, Christiansen and Delgado (2017) studied continuous and discrete-time models for short-sea inventory routing problem, similar to the problem presented in Agra, Christiansen and Delgado (2013), considering dedicated compartments for each product. They first present a discrete formulation based on the model of Agra, Christiansen and Delgado (2013) where consumption rates vary. This model was adapted for constant consumption rates, and two sets of constraints were added to ensure that inventory bounds within the time period are preserved, making it more similar to the continuous-time model after described. Either the discrete and continuous-time formulations are extended (arc load flow and multi-commodity reformulation). The formulations were improved using valid inequalities to impose a minimum number of visits and tightening constraints on time linking and maximum amounts to be loaded. Comparisons between the models were made considering size, duality gap, and processing time. Results demonstrated that discrete-time formulations have better integrality gaps, while continuous-time formulations usually are solved faster.

Hemmati et al. (2016) presented a MIRP focused on short-sea applications with a continuous-time formulation. In the problem, each product can be consumed or produced in any port, and their inventories are considered separately, where each port has different storage capacities for each product type. Also, different inventories' limits are considered at the end of the time horizon. On the other hand, vessels can carry any product, and load/unload in any order, but the allocation of products to compartments was not considered. The problem assumes that vessels start and end their routes empty. A two-phase matheuristic called Hybrid Cargo Generating and Routing (HCGR) was proposed to solve the problem. It is based on the idea of Hemmati et al. (2015) to convert an inventory routing problem to a ship routing problem. In the first phase, cargoes are created based on demand, production, consumption rates of the ports, defining time windows based on the inventory capacities. After, an adaptive large neighborhood search was used to solve the ship routing problem. The HCGR iteratively changes the generated cargoes based on the information obtained in solving the second phase. They compare the HCGR with the CPLEX solver on a small instance set and evaluate the HCGR on the set with large and more realistic instances.

The work of Stanzani et al. (2018) extended the works of Al-Khayyal and Hwang (2007), Christiansen (1999) by modeling a MIRP for pickup and delivery of different types of oil from offshore platforms to onshore terminals of the Brazilian company Petrobras. The inventory management was considered only in the pickup points (platforms), which produces a unique and specific oil type. For the terminals, it was assumed that they have a large capacity within the planning horizon. Specific constraints were considered, such as the use of dynamic positioning systems that allow vessels to dock at platforms with higher levels of load onboard. A multi-start heuristic was used for building feasible solutions of pickup and delivery routes, followed by local search procedures that aim to reduce the number of docks at the platform/terminal. The computational experiments were carried out on real-based instances and demonstrated that the proposed approach is suitable for the company's real operations.

In Agra et al. (2015), a stochastic version of the problem described by Agra, Christiansen and Delgado (2013), Agra et al. (2014) was presented. They considered uncertainty in sailing and port times. Inventory constraints were only accounted for consumption tanks at ports. The problem is formulated as a two-stage stochastic programming with recourse and solved by decomposition on a master problem and one subproblem for each scenario, which derives optimality cuts. Computational results were carried out on ten real-world instances, showing the method's effectiveness and the importance of using stochastic programming instead of a deterministic scenario.

2.2.2 Variable Production/Consumption Rates at Ports

The works presented in this section have variable production and consumption rates along the time horizon.

2.2.2.1 Single-product

A MILP integrated into a decision support tool was proposed by Furman et al. (2011) for the transportation of vacuum gas oil from Europe to the United States. The problem considers several side constraints, such as draft limits in a subset of ports and minimum transport cargo for each vessel. Also, additional vessels can be chartered for performing a unique voyage (from a loading port to a discharging port) if necessary. The work presented a description of instances and how the costs were calculated. The problem was solved using a mathematical solver, which takes on average 10 to 15 minutes to obtain a solution. However, no specific computational results were presented.

In a later work, Song and Furman (2013) proposed a modeling framework for the inventory routing problem, which can accommodate practical features. The model was an enhancement of the system introduced by Furman et al. (2011). A study case on a MIRP was made to evaluate the model efficiency. The solution method applies a preprocessing phase to reduce the search space, followed by a branch-and-cut with branching strategies and valid inequalities. A large neighborhood search was used to improve the solution by fixing variables from a feasible solution and solving subproblems iteratively. Ten real-based generated instances were solved until optimality, ensuring the effectiveness of the proposed framework.

Engineer et al. (2012) proposed a branch-and-cut-and-price algorithm for the problem described in Furman et al. (2011), Song and Furman (2013). The master problem considers the inventory balance on ports, while the subproblem is the fixed-charge shortest-path problem, which can be solved to optimality by dynamic programming labeling algorithm. Three families of cuts and four branching strategies were proposed. The computational experiments used nine instance classes varying the number of vessels and the number of supply/demand ports. They analyze the impact of the cuts on the master problem. The BCP was compared with the compact arc-flow formulation of Song and

Furman (2013), showing the approach's superiority considering either time, bounds, and solution quality. For the same problem of Song and Furman (2013), a branch-and-price guided search (BPGS) is presented in Hewitt et al. (2013). The BPGS uses an extended formulation which restricts the original problem through a set of constraints obtained by a pricing problem. Six local search schemes based on the problem structure were proposed for improving the solution, being solved in parallel. The algorithm was compared with the CPLEX solver, producing high-quality solutions in a short processing time.

The problem of transportation of crude oil from a central supplier with unlimited capacity to several customers was presented by Shen, Chu and Chen (2011). The transport can be directed from the supplier to the customer harbor only using tankers through a general route or canals, or it can be intermediated by input and output ports, which are connected by pipelines. Then the oil from the output port is transported to the customer by tankers. A Lagrangian relaxation with valid inequalities was proposed to find near-optimal solutions. It was compared to a previously proposed metaheuristic and a MILP solver. The efficiency of the method was also evaluated with a problem variant allowing partially loaded tankers.

Rocha, Grossmann and Aragão (2013) considered a MIRP where tankers transport oil from offshore platforms to an onshore terminal. They assume that tankers must be loaded to the full capacity at platforms, performing only direct voyages. Also, an unlimited fleet of vessels is given, as they considered that vessels could be chartered. The initial model was reformulated through the inventory constraints, and new cascading knapsack inequalities were proposed for solving instances where only the initial formulation is not sufficient. Using the same problem of Rocha, Grossmann and Aragão (2013), Aizemberg et al. (2014) devised six formulations of the problem, including a Dantzig-Wolfe decomposition, which was solved by a column generation based heuristic. The computational analysis demonstrated that the new accumulated rounded cascading formulation performed better on obtaining optimal solutions against the other models. Also, the Dantzig-Wolfe model provided tighter bounds. Finally, tests allowing limited inventory violations demonstrated that better solutions (originally infeasible) could be obtained.

The work of Agra et al. (2013) considered a short-sea MIRP, which assumes that a vessel cannot wait in a port after finished its operation, i.e., it can wait before starting to operate. However, once it began to load or discharge the product, the operation needs to be continuous, and after finished, the vessel must travel to another port or complete its route. A demurrage cost is paid while a vessel waits in a port before operating, and vessels can be idle along the entire planning horizon. A port can have one or more berths that vessels can operate simultaneously. Finally, a maximum amount can be loaded or discharged at a port by a vessel. If a vessel wants to load/discharge more than this maximum amount, it will need to operate two time periods consecutively. They proposed a discrete-time fixed-charge network flow model (FCNF), and new valid inequalities generalized from the lot-sizing problem were developed with branching priorities. The proposed model obtained tighter bounds and solved the problem instances faster than the standard time-space network.

Papageorgiou et al. (2014a) proposed an iterative algorithm with two phases. They considered a realistic assumption that one vessel must visit at most two ports of the same type and geographical region sequentially. A special time-space network was built to incorporate the assumption indirectly. The model was improved with valid inequalities, and an improvement phase with a MIP-based local search was applied. The proposed approach provided high-quality solutions in a reasonable processing time.

Papageorgiou et al. (2015) proposed an Approximate Dynamic Programming (ADP) framework for solving a simple MIRP with long planning horizons (up to 360 days). The ports' inventory was considered soft constraints, while vessel inventory was not tracked. The model assumes that a vessel only performs voyages from a loading port to a discharging port at full capacity and from discharging ports to a loading port only if it is empty. The ADP approach was able to find good solutions faster than a commercial solver with emphasis on feasible solutions with local search procedures. For the same problem, Papageorgiou et al. (2018) presented different matheuristic and hybrid solution approaches. They investigated constructive methods such as the rolling horizon heuristic combined with different problem-specific assumptions and improvement methods, such as k-opt local search. Several computational experiments were performed, comparing different solvers' parametrization to the various algorithms presented. The matheuristic approaches found new best know solutions for 26 instances.

The work of Diz, Oliveira and Hamacher (2017) used the model proposed by Agra et al. (2013) for solving a crude oil and supply problem faced by the vertically integrated oil and gas company Petrobras. In this problem, the vessels are responsible for collect oil from offshore platforms and transport it to onshore terminals. Six different scenarios were generated from one reference instance for testing purposes. The model was compared with the manual routing and schedule performed at the company, obtaining, on average, 20% of cost reductions.

Munguía et al. (2019) explored the use of the parallel alternating criteria search (PACS) for solving the MIRP instances proposed by Papageorgiou et al. (2014b). PACS is a general-purpose algorithm that consists of a parallel primal heuristic based on a large neighborhood search, which solves restricted sub-MIPs of the original problem. The work demonstrated the PACS effectiveness for solving MIRPs, whose performance was improved when specific problem domain of the MIRP was prototyped into the algorithm framework, such as specific objective penalizations and variable fixing schemas.

Works concerning the MIRP variation in which the transported product is the liquefied natural gas usually consider either variable production and a single product. The first work in the liquefied natural gas inventory routing problem was proposed by Grønhaug and Christiansen (2009). They considered that vessels must be fully loaded but can unload a variable number of tanks at each discharging terminal, such that at most two consecutive visits at discharging ports can be performed. In this problem, the production and demand at ports are bounded variables that should be defined. Arc and path flow formulations were presented and compared. A path generator is proposed for enumerating all possible columns in the second model. However, only small instances were solved to optimality.

A branch-and-price algorithm was proposed in Grønhaug et al. (2010) for the LNG-IRP presented in Grønhaug and Christiansen (2009). An ad hoc dynamic programming algorithm solves the subproblems while branching strategies and accelerating techniques were proposed for the column generator and B&P algorithm. The computational results demonstrated that the B&P algorithm was quite faster than the enumeration algorithm of Grønhaug and Christiansen (2009). However, the optimality of the solutions could not be proved in the specified time limit. The work of Andersson, Christiansen and Desaulniers (2016) proposed a new formulation for the LNG-IRP model of Grønhaug et al. (2010). It consists of splitting the vessel path in duties, defined as a sequence of ports starting in a liquefaction plant (loading port) and one or two regasification terminals, ending again at a liquefaction plant. Vessels paths are pre-generated as the number of duties is limited, and a branch-and-bound algorithm solves the model. The new model provided a tight formulation and could be improved using problem-specific valid inequalities, outperforming previous results obtaining optimal and new best-known solutions.

A planning level for designing a supply chain of LNG is presented in Goel et al. (2012). In this problem, the production rates are given as data, while the consumption (also called regasification) rates at discharging ports are decision variables. Besides considering seasonal traveling times, the problem ensures that vessels can only navigate at full capacity or empty, i.e., no split pickup or deliveries are allowed. The inventory level constraints consider lost production and stock-out by using slack variables penalized in the objective function. A constructive greedy heuristic was used for building an initial solution, which was improved by two large neighborhood search methods. The first method was based on delaying departure voyages, and the second consist of improving the routes of only two vessels iteratively, like a 2-opt neighborhood. Computational results compared the algorithm's performance against the CPLEX solver, considering different selection strategies for the two vessel improvement routine.

The work of Fodstad et al. (2010) proposed a rich LNG-IRP, extending the problem presented in Grønhaug et al. (2010) by addressing a larger part of the LNG supply chain, considering importation and exportation of LNG, natural gas hub, among others. Vessels can become unavailable due to maintenance, and additional vessels can be chartered. The model was evaluated with new proposed instances, presenting the benefits of considering rich formulations for the LNG planning problem. An extended fix-and-relax matheuristic was used in Uggen, Fodstad and Nørstebø (2013) for solving the rich model of Fodstad et al. (2010). The method divides the planning horizon into subproblems with short time horizons, solving them iteratively. For reducing the size of the subproblems, a horizon cut and light model strategies were used. A fix and optimize algorithm that also divided the planning horizon into intervals was used for improving the solution. Several computational tests were performed to compare the standard fix and relax with the extended version and the contribution of the MIP-based local search.

For dealing with larger instances, Goel et al. (2015) developed two constraint programming strategies for the problem proposed in Goel et al. (2012), considering simplifications such as no seasonal traveling times and variable consumption rate. The computational results demonstrated the method's superiority against the large neighborhood search proposed in Goel et al. (2012). Mutlu et al. (2016) developed a model for an annual delivery program for liquefied natural gas where the problem should define the production rate. It allows split delivery and soft time windows constraints, which were ignored in other works. A vessel routing heuristic builds multiple solutions that were used for a warming start of the mathematical solver. Computational tests performed on available author instances demonstrated that allowing split deliveries may lead to significant cost reductions.

2.2.2.2 Multi-products MIRPs

A MIRP involving the transportation of several liquid bulks was proposed in Ronen (2002). The problem considers that vessels are homogeneous and chartered and assumes that a vessel only visits one loading port before visiting a discharging port and vice-versa. Also, penalization costs by shortfalls of safety stock were incurred. Data derived from the distribution of products from refineries was used to evaluate the model, which solved small instances to optimality. For large instances, more processing time was needed, and the solutions' optimality was not proved. Thus, a heuristic approach based on the logic of the planner was proposed with an improvement phase. Although obtained a solution usually less than a second, the heuristic performance was inferior to the CPLEX running within five minutes limit.

A problem faced by the shipment planning of a refinery company in Sweden was presented in Persson and Göthe-Lundgren (2005), where bitumen products are transported from refineries to depots. They considered that ships only visit at most two depots after departing from a refinery and assume that the products can be mixable. The objective aims to minimize different costs associated with the production, transportation, inventory holding, deviation from inventories target, and violation of inventories constraints. Arc-flow and path flow models were proposed with valid inequalities to ensure a minimum number of visits at each port. A column generation algorithm with a restricted tree search and fixing strategies was used for finding integer solutions in the path flow formulation. Computational experiments carried out on instances based on real company operations evaluate the linear relaxation and integer solution quality of the two formulations.

The work of Andersson (2011) presented a pulp distribution problem, where different products are transported from pulp mills to terminals. No capacity at terminals was considered, but limited operation quantities per time period should be respected. If necessary, spot vessels can be chartered for a single trip, i.e., pulp transport from one pulp mill to one terminal. A path flow model was proposed with a branch-and-price algorithm used as a decision support tool.

A decision support tool for the distribution of calcium carbonate slurry was presented in Dauzère-Pérès et al. (2007). One production port is responsible for supplying several consumption ports in Europe. A MILP was proposed, where only small instances could be solved. For dealing with large instances, a memetic algorithm was proposed, and quantitative benefits were presented. The work of Rakke et al. (2011) proposed a rolling horizon heuristic for an annual delivery program of LNG-IRP. Two types of liquefied natural gas were considered, and the inventory is tracked only at the production port, while long-term contracts dictate the amount of LNG to be delivered to clients every month or in the entire year. Penalties were incurred for under and over-deliver of LNG at contracts. The ships can only carry one type of LNG at a time, and they should be unavailable after a certain period of activities for maintenance. Besides the rolling horizon strategy, a solution space reduction was developed to decrease the number of symmetrical solutions. A heuristic improves the solution by using additional variables and constraints for limiting the search space. The proposed algorithm found better solutions than a heuristic proposed in earlier work, although being more time-consuming.

Bilgen and Ozkarahan (2007) described a distribution problem faced by a company that manages wheat distribution planning. In this MIRP variant, different products are available at loading ports, and clients can demand pure products or a blend of products. An unlimited number of different vessels are available to load products on at most two loading ports and discharges all cargo at one client. The problem assumes that vessels have sufficient compartments for storage of the different products. In this particular problem, the inventory at ports was not considered, but each customer's demand must be attended to in each time-period (month). The proposed MILP model aims to minimize blending, loading, transportation, and inventory costs.

A study case on a cement industry in Norway was presented by Christiansen et al. (2011). Different non-mixable cement products are transported in separate compartments on vessels, while production and consumption ports have dedicated silos for each product with lower and upper limits. As the fixed fleet is not sufficient to supply peak demand seasons, priorities for each silo are given, and the problem should decide which demands to supply. A constructive heuristic was proposed, while a genetic algorithm was responsible for evaluating the heuristic parameters for obtaining better solutions.

A sustainable MIRP is described in De et al. (2017) for the transport of different container types. The problem considers low-steaming policies, where the vessel's speed must be decided, given rise to a mixed-integer nonlinear programming model. They considered that vessels consume heavy fuel oil while they are sailing and marine diesel oil while at a port. These fuels have different prices. Time windows are considered at ports but can be violated if the vessel arrives before starting the time window, or the vessel finishes its operation after the end of the time window. A particle swarm optimization for composite particle (PSO-CP) was developed, and their results considering fuel costs were compared with basic particle swarm optimization and genetic algorithm.

A multi-product MIRP with undedicated compartments was proposed by Foss et al. (2016). They extend the model of Agra et al. (2013), and valid inequalities based on the problem structure and tightening constraints were proposed. Unlike Siswanto, Essam and Sarker (2011), a ship was allowed to return to a loading port before becoming empty. Computational experiments were carried out on instances containing vessels with up to three compartments, comparing the model's performance with and without the valid inequalities. A solution analysis demonstrated that although undedicated compartments increase the problem complexity, it can provide potential economic savings if compared with problems that do not consider compartments or consider dedicated compartments.

2.3 Summary Tables

The works cited in Section 2.2 are summarized in Table 2.1 for the single-product MIRPs, and in Table 2.2 for multi-products MIRPs. Column *P/C rate* refers to the production or/and consumption rates at loading and discharging ports. Column *Time* informs if the mathematical model consider continuous time (**C**) or discrete time (**D**) formulations. Column *Nav. type* refers to the type of navigation considered: deep-sea (**DS**) or short-sea (**SS**). Column *Oper. bounds* report if there is a lower (**Min**) or/and upper (**Max**) limit on the amount of product(s) that can be loaded or unloaded at a port in each visit or time period. The upper limit does not consider the implicit value of the vessel capacity. Column *Bert cap.* informs the berth capacity at ports, which can be one (**1**) or multiple ($\geq$ 1). Column *Soft constr.* informs if the work considers soft constraints with slack variables. They can be on the time windows (*TW*) or on the port inventories (**Port inv.**). When slack variables on port inventories are bounded, the notation becomes $\overline{\textbf{Port inv.}}$. Column *Draft limit* indicates if draft limits of vessels are explicitly considered on the model, i.e., by specific draft limit constraints. Column *Method* refers to the solving method used. **MIP-LS**: MIP-based local search, **ADP**: Adaptive dynamic programming, **Path Gen.**: path generation, **LNS**: large neighborhood search, **CG**: column generation, **CGH**: column generation based heuristic, **CP**: constraint programming, **BB**: branch-and-bound algorithm, **BP**: branch-and-price algorithm, **BCP**: branch-and-cut-and-price algorithm, **BPGS**: branch-and-price guided search, **GA**: genetic algorithm, **ILS**: iterated local search, **LS**: local search, **RHH**: rolling horizon heuristic, **Memetic**: memetic algorithm, **PSO-SP**: Particle swarm optimization for composite particle. **PACS**: Paralellel Alternating Criteria Search. Columns *B* and *VI* indicate if the work proposed branching strategies and valid inequalities for the proposed model, respectively. The remaining columns refers to the maximum size of the instances, where $|\mathcal{V}|$ is the number of vessels, $|\mathcal{J}|$ is the total number of ports, and $|\mathcal{T}|$ refers to the size of the planning horizon, which is measured according to the information in column *Unit*: Hour (**h**), day(**d**), month(**m**). Note that the instance size considers the major number of each attribute, even if such an instance does not exist. Especially for Table 2.2, column $|\mathcal{K}|$ refers to the number of products, while column *Port tank* reports how the inventory of different products are considered at ports: if each product has it own inventory with lower and upper limit (**Dedicated**), or if it is not considered. Similarly, column *Vessel compart.* report if vessels compartment are considered, being them **Dedicated** or **Undedicated** for each product type. When a field is empty for one or more columns, the information was not provided by the correspondent work, or the feature was not considered in the specific problem.

Table 2.1: Works on single-product Maritime Inventory Routing Problem.

Work	*P/C rate*	*Time*	*Nav.Type*	*Oper. Bounds*	*Berth cap*	*Soft constr.*	*Draft limits*	*Method*	*B*	*VI*	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{J}\rvert$	$\lvert\mathcal{T}\rvert$	*Unit*
Grønhaug and Christiansen (2009)	Variable	D	DS	Min/Max	≥ 1			Path Gen. + BB			5	6	60	d
Grønhaug et al. (2010)	Variable	D	DS	Min/Max	≥ 1			BP			5	6	60	d
Fodstad et al. (2010)	Variable	D	DS	Min/Max	≥ 1			Solver			8	7	181	d
Furman et al. (2011)	Variable	D	DS	Min/Max	1	$\overline{\text{Port inv.}}$	•	Solver						d
Shen, Chu and Chen (2011)	Variable	D	DS		≥ 1			Lagrangian relaxation		•		11	12	m
Goel et al. (2012)	Variable	D	DS	Min/Max	≥ 1	Port inv.		Heuristic + LNS			69	11	360	d
Engineer et al. (2012)	Variable	D	DS	Min/Max	1		•	BCP	•	•	6	10	60	d
Agra et al. (2013)	Variable	D	SS	Max	≥ 1			Solver	•	•	5	6	60	12h
Song and Furman (2013)	Variable	D	DS	Min/Max	1		•	BB + LNS	•	•	8	10	60	d
Hewitt et al. (2013)	Variable	D	DS	Min/Max	1		•	BPGS			6	10	45	d
Uggen, Fodstad and Nørstebø (2013)	Variable	D	DS	Min/Max	≥ 1			Fix-and-relax			8	10	180	d
Rocha, Grossmann and Aragão (2013)	Variable	D	SS					Solver		•		11	30	d
Papageorgiou et al. (2014a)	Variable	D	DS	Min/Max	≥ 1	$\overline{\text{Port inv.}}$		Two-stage decomposition	•	•	17	13	60	d
Papageorgiou et al. (2015)	Variable	D	DS		≥ 1	Port inv.		ADP			70	13	360	d
Aizemberg et al. (2014)	Variable	D	SS					Solver, CGH				11	30	d
Goel et al. (2015)	Variable	D	DS	Min/Max	≥ 1	Port inv.		CP			69	11	360	d
Zhang et al. (2015)	Constant	D	DS		≥ 1			Two-phase heuristic			8	12	60	d
Agra et al. (2015)	Constant	C	SS			Port inv.		L-shaped like decomposition		•	4	9	8	d
Agra et al. (2016)	Constant	C	SS	Min/Max	1	Port inv.		MIP-LS			5	6	30	d
Agra et al. (2017)	Constant	D	SS	Min	1	Port inv.		Matheuristic		•	2	61	10	d
Andersson, Christiansen and Desaulniers (2016)	Variable	D	DS	Min	≥ 1			Path Gen + BB		•	5	6	75	d
Mutlu et al. (2016)	Variable	D	DS		≥ 1	Ports inv./TW		Solver, Vessel Route Heuristic		•	40	17	360	d
Diz, Oliveira and Hamacher (2017)	Variable	D	SS	Min/Max	≥ 1			Solver			11	11	15	d
Papageorgiou et al. (2018)	Variable	D	DS		≥ 1	Port inv.		RHH			70	13	360	d
Agra et al. (2018)	Constant	C	SS	Min/Max				Decomposition, ILS		•	5	6	30	d
Munguía et al. (2019)	Variable	D	DS	Min/Max	≥ 1	$\overline{\text{Port inv.}}$		PACS			70	13	360	d
Rodrigues et al. (2019)	Constant	C	SS	Min/Max				Solver		•	5	6	30	d
This Thesis	Variable	D	DS	Min/Max	≥ 1	$\overline{\text{Port inv.}}$		R&F, F&O, Multi-start, LNS		•	17	13	60	d

Source: From the author (2020).

Table 2.2: Works on multi-product Maritime Inventory Routing Problem.

Work	*P/C rate*	*Time*	*Nav. Type*	*Oper. Bounds*	*Berth cap*	*Soft constr.*	*Draft limits*	*Method*	*B*	*VI*	$\|\mathcal{V}\|$	$\|\mathcal{J}\|$	$\|\mathcal{T}\|$	*Unit*	$\|\mathcal{K}\|$	*Port tank*	*Vessel compart*
Ronen (2002)	Variable	D		Min/Max		Port inv.		Solver				7	30	d	5	Dedicated	
Persson and Göthe-Lundgren (2005)	Variable	D	SS			Port inv.		CG		•	3	18	42	6h	4	Dedicated	
Al-Khayyal and Hwang (2007)	Constant	D	SS		≥ 1			Solver			4	4	10	d	3	Dedicated	Dedicated
Bilgen and Ozkarahan (2007)		D					•	Solver			∞	6	3	m	8		
Dauzère-Pérès et al. (2007)	Variable	D	SS	Min				Memetic			17	11	84	d	16	Dedicated	
Li, Karimi and Srinivasan (2010)	Constant	C	DS		≥ 1			Solver			5	8	80	d	2	Dedicated	Dedicated
Andersson (2011)	Variable	D	SS	Max				BP			+3	27	147	8h	30	Dedicated	
Christiansen et al. (2011)	Variable		SS		1		•	GA			5	61	14	d	5	Dedicated	Undedicated
Siswanto, Essam and Sarker (2011)	Constant	C			≥ 1			Heuristic			3	4	15	d	2	Dedicated	Undedicated
Rakke et al. (2011)	Variable	D	DS		≥ 1	Port inv.		RHH			46		366	d	2	Dedicated	
Agra, Christiansen and Delgado (2013)	Constant	D+C	SS		1	TW		Solver		•	2	7	12	d	4	Dedicated	
Agra et al. (2014)	Constant	C	SS	Min/Max	1			RHH, matheuristics		•	2	7	180	d	4	Dedicated	Dedicated
Hemmati et al. (2016)	Constant	C	SS	Min/Max	1			Two-phase matheuristic			7	24	1440	h	3	Dedicated	
Agra, Christiansen and Delgado (2017)	Constant	D,C	SS	Min/Max	1			Solver		•	2	7	15	d	4	Dedicated	Dedicated
Foss et al. (2016)	Variable	D	SS	Max	≥ 1			Solver		•	4	8		d	4	Dedicated	Undedicated
De et al. (2017)	Variable	D			≥ 1	TW		PSO-CP			6	10	22	d	2	Dedicated	
Stanzani et al. (2018)	Constant	C	SS	Min	≥ 1		•	Multistart+LS			25	18	773	h	13		

Source: From the author (2020).

2.4 Solution Methods

In this section, some background on the proposed solution methods is presented.

2.4.1 Relax-and-Fix Algorithm

The relax-and-fix algorithm was first described in Dillenberger et al. (1994) for a lot-sizing problem. In that work, it was named as a fix-and-relax algorithm, although it is the same as the R&F framework, dividing the time horizon into stage periods or intervals. At the first iteration, all integer variables are relaxed, except those belonging to the first stage. The problem is then solved, followed by fixing the first-stage variables with the obtained values, while the integrality constraint is reintroduced on variables of the next stage. The new subproblem is then solved, and the procedure repeats until all stages have the integrality constraints reintroduced. Wolsey (1998), Pochet and Wolsey (2006) defines R&F in a generic form, splitting the integer variables into disjoint sets according to their level of importance in the problem, such that the integrality constraint is re-inserted first in the variables with more importance. The pseudo-code of the basic R&F is described in Algorithm 1.

Algorithm 1: Relax-and-fix algorithm.

$Q^1, \ldots, Q^R \leftarrow R$ disjoint sets containing the integer variables of the MIP
Remove the integrality constraints of all variables
$i = 1$
while $i \leq R$ **do**
    Re-insert the integrality constraints of set Q^i
    Solve the MIP containing all sets
    Fix the variables of set Q^i with the obtained solution values
    $i = i + 1$
end while

Usually, the decomposition is made based on the time horizon as in Dillenberger et al. (1994), where the "most important" variables are associated with the beginning of the planning horizon. Such time decomposition is also refereed as rolling horizon heuristic (RHH) (RAKKE et al., 2011; AGRA et al., 2014; PAPAGEORGIOU et al., 2018). However, a difference between R&F and RHH is that R&F initially considers all planning horizons, while RHH considers a part of the planning horizon, in which integer variables are not relaxed.

The R&F is usually used in production planning and scheduling problems. Stadtler (2003), Federgruen, Meissner and Tzur (2007) used a time decomposition for multi-items lot-sizing problems. Stadtler (2003) considered rolling schedules where the problem is divided into lot-sizing time windows. These time windows have three parameters: number of periods, number of overlapping periods, and relaxed periods. Thus, the algorithm consists of solving each time window iteratively. Federgruen, Meissner and Tzur (2007) tested the algorithm called progressive interval heuristic with and without overlaps between the time intervals. Overlap defines that only a portion of the interval previously solved with the integrality constraints is fixed. The fixing of variables is also accounted for the original continuous variables. Results revealed that when overlaps were not considered, the algorithm tends to be faster. However, the solution quality is higher when considering overlaps. Beraldi et al. (2008) used two rolling horizon heuristics for solving a lot-sizing problem with identical parallel machines. Also, fix-and-relax heuristics based on time, product, and hybrid decompositions were proposed. The computational studies showed that the fix and relax provided better solutions than the rolling horizon heuristic because the RHH delays as much as possible the decisions to meet the demand restrictions.

For a soft drink lot-sizing problem, Ferreira, Morabito and Rangel (2009) tested three decomposition approaches with different fixing strategies. The decompositions were based on machines, macro, and micro time periods. The work of Akartunalı and Miller (2009) proposed a relax-and-fix algorithm and an LP and fix to generate several feasible solutions in a heuristic framework for solving a production planning problem. LP-and-fix consists of solving the problem relaxation and verify if some integer variables assume integer values. These variables are then fixed, and the problem is solved again. They discuss which binary variables should be fixed and observed no relevant difference between fixing only those that take value 1 or only those that take value 0. In Mohammadi et al. (2010), rolling horizon and fix and relax strategies were proposed for a multi-product and multi-level capacitated lot-sizing problem. They differ from considering simplifications on the relaxed interval, such as eliminating variables, simplifications on the integer interval to give an upper bound for the original problem, and the use of additional constraints for restricting the problem. Uggen, Fodstad and Nørstebø (2013) tested simplification strategies and a horizon cut for the end of the planning horizon, reducing the computational complexity and time-consuming.

Different approaches can be used for solving each subproblem in the R&F framework. In Pochet and Warichet (2008), if a solution becomes infeasible at some iteration due to the variables fixing, a relaxation is solved by unfixing the previously fixed variables and allowing just a limited number of them to change their value, like a local branching strategy. Combining a rolling horizon model with the relax and fix, Araujo, Arenales and Clark (2007) tested three heuristic methods for solving the first subproblem. They vary between using a descent heuristic with and without a neighborhood diminishing and a simulated annealing strategy. Agra et al. (2014) tested the use of local branching and feasibility pump heuristic with a special rounding scheme for solving the subproblems in their rolling horizon framework.

2.4.2 Fix-and-optimize algorithm

The fix-and-optimize algorithm (F&O) is used in improvement procedures, starting from an integer solution. It consists of dividing the integer variables into L disjoint sets according to some criteria, similar to the R&F. Such sets can be defined using different strategies. Initially, it fixes all integer variables with the feasible solution's values and iteratively allows a set of variables to be optimized. A pseudo-code of the F&O is described in Algorithm 2.

Algorithm 2: Fix-and-optimize algorithm.

$Q^1, \dots, Q^L \leftarrow L$ disjoint sets containing the integer variables of the MIP
$S \leftarrow$ feasible solution of the problem
Fix all integer variables according to the S values
$i = 1$
while <Some criteria> **do**
    for i to L **do**
        Unfix variables of set Q^i
        $S' \leftarrow$ solve the MIP containing all sets
        Fix the variables of set Q^i with the values of S'
        $i = i + 1$
    end for
end while

In the MIRP literature, we can find different names for algorithms that are equivalent to the F&O. For example, Goel et al. (2012), Song and Furman (2013) used the term *large neighborhood search* for improving a MIRP solution by fixing the variables associated with all vessels, iteratively selecting two vessels to be optimized. Taillard and Voss (2002) considered the name *POPMUSIC* (Partial Optimization Metaheuristic Under Special Intensifications and Conditions) for a general framework that divides the problem into smaller parts. Starting from a feasible solution, it fixes the variables and selects a seed part to be optimized at random. A second part is also selected according to a distance function from the seed part. Hewitt et al. (2013) used the term *local search schemes*, assuming that part of the solution is known, and the unknown part can be solved until optimality. Each scheme consists of fixing a subset of variables while improving the remainder subsets. The sets of variables are defined according to supply ports, demand ports, vessels, and time windows. In Agra et al. (2017) the term considered was *local search matheuristic* that solves a single-vessel route iteratively by using a branch-and-cut algorithm for short processing time. Papageorgiou et al. (2014a) used the term a *MIP-based local search strategy*, which is based on the named *fixing supply and fixing demand strategy* proposed by Hewitt et al. (2013).

The F&O was also used on other problems, such as the lot-sizing problem. The work of Lang and Shen (2011) proposed F&O strategies for a capacitated lot-sizing that relies on time and product decompositions. Toledo et al. (2015) also used R&F and F&O for solving a multi-level lot-sizing problem. They considered a matrix where each entry corresponds to an integer variable of the problem associated with the time horizon. From this matrix, three strategies were defined for the R&F: row-wise, column-wise, and value-wise. The last one re-insert the integrality constraint at a limited number of variables that have their fractional value closest to 0.5. For the F&O strategy, only row and column-wise were used considering the overlap between the matrix entries, i.e., a certain percentage of the variables are re-optimized in two consecutive iterations.

2.4.3 Multi-start Algorithm

Multi-start metaheuristics are useful in solving hard combinatorial optimization problems where it is difficult to define an efficient and effective neighborhood structure for using local search methods to improve a unique generated solution Martí, Moreno-Vega and Duarte (2010). The basic algorithm consists of generating a set of solutions from random starting points to obtain diversification. Optionally, improvement procedures can be applied to each solution generated. For a comprehensive review of multi-start methods, we recommend the work of Martí, Moreno-Vega and Duarte (2010).

2.4.4 Large Neighborhood Search

The large neighborhood search (LNS) method improves one or more solutions obtained by the multi-start algorithm. This method explores a sample of a solution neighborhood, as exploring the whole neighborhood is not practical. The LNS iteratively uses destroy and repair methods to improve the solution gradually. The destroy method removes part of the solution (usually turning it infeasible), while the repair method is responsible for rebuilding the solution in a different way. It is equivalent to the fix-and-optimize algorithm, where the destroy method is equivalent to fixing a subset of the model variables, while the repair method corresponds to solving the subproblem. The difference between F&O and LNS is that the first solves the subproblems using a mathematical solver, while the second solve them heuristically. For a survey on LNS, we refer to the work of Pisinger and Ropke (2010).

3 PROBLEM DESCRIPTION

This chapter describes a variant of the MIRP studied in this work. It is presented a time-space network (TS) model followed by its reformulation in a fixed-charge network flow (FCNF) model derived from the literature. Finally, formulations improvements such as valid inequalities, additional constraints, and flow variables tightening are presented. They are referred to as “formulation components" throughout the thesis.

The MIRP considered in this work is the same problem described in Papageorgiou et al. (2014a), which is an extension of the core-model proposed by Papageorgiou et al. (2014b). In this discrete-time problem, a single product is produced at different production (or loading) ports and consumed at consumption (or discharging) ports. The loading and discharging ports are grouped separately in one or more geographical regions. The problem is considered a deep-sea MIRP, where the traveling times are much larger than the operating times, characterizing intercontinental trips. The ports' production and consumption rates are variable along the planning horizon, and minimum and maximum inventory bounds are imposed for each port and upper and lower limits on loading or discharging quantities per time period. The number of available berths per port imposes the maximum number of vessels that can operate in each port simultaneously. The product is transported by different classes of vessels, which can differ in capacity, traveling costs, and time. Each vessel starts its voyage from a dummy source node to its initial port. The initial port and the time in which the vessel becomes available are given as input parameters. When arriving at a port, a vessel can start to operate, occupying a berth, or can wait outside the port without operating. After finishing the operation, it can depart to another port or can end its route, arriving at a dummy sink node. Vessels must be at full capacity when traveling from a loading port to a discharging port, and they can only return to a loading port after discharging all cargo on board, being empty. The initial inventory at ports and the initial load on board the vessel are known in advance. As maritime transportation usually involves uncertainty on data, and inventory bounds at ports are very conservative constraints, it is possible to consider penalizing the bound violation at the ports' inventory, using soft constraints. Their use allows a more robust solution if it is impossible for any vessel to arrive at a port before its inventory violation (PAPAGEORGIOU et al., 2014b). Thus, the problem assumes that a limited quantity of product can be bought from or sold to a simplified spot market along the planning horizon. The problem consists of defining each vessel's route and schedule, besides the amount loaded and discharged at ports to maximize the revenue obtained by the quantity discharged at discharging ports, subtracting the traveling and operations costs and penalties from the ports inventory violation.

The next sections present two discrete-time formulations for the MIRP. The first is a time-space network presented by Papageorgiou et al. (2014b), Papageorgiou et al. (2014a). The second reformulates the time-space network, transforming it in a fixed-charge network flow model and is based on the work of Agra et al. (2013).

3.1 Time-space Network Model

The TS model for a MIRP was first introduced by Song and Furman (2013). Let $\mathcal{V}$ be the set of vessels, $\mathcal{J}$ the set of ports, and $\mathcal{T} = \{1, \ldots, |\mathcal{T}|\}$ the set of time periods. Ports are split into subsets $\mathcal{J}^{\mathrm{P}}$ for loading ports, and $\mathcal{J}^{\mathrm{C}}$ for discharging ports, where $\mathcal{J} = \mathcal{J}^{\mathrm{P}} \cup \mathcal{J}^{\mathrm{C}}$, and $\mathcal{J}^{\mathrm{P}} \cap \mathcal{J}^{\mathrm{C}} = \emptyset$. Ports are grouped in production regions $\mathcal{R}^{\mathrm{P}}$ and discharging regions $\mathcal{R}^{\mathrm{C}}$, such that $\mathcal{R} = \mathcal{R}^{\mathrm{P}} \cup \mathcal{R}^{\mathrm{C}}$. Each vessel $v \in \mathcal{V}$ has its own arc set, while the nodes set is shared by all vessels. Regular port-time nodes $n = (j, t) \in \mathcal{N} = \mathcal{J} \times \mathcal{T}, j \in \mathcal{J}, t \in \mathcal{T}$ define where the events (such as loading or discharging operations by a vessel) take place. A dummy source node $o(v)$ represents the starting of vessel v route, while a dummy sink node $d(v)$ denotes the end of its route. A vessel route is the sequence of port visits along the planning horizon between the source node and the sink node. Figure 3.1 illustrates the time-space network model and a possible route for a vessel v between two ports i and j.

In Figure 3.1, vessel v starts its voyage departing from source node $o(v)$ to port j, arriving at time period $t = 2$. The arc linking the source node and the first port is fixed and given as an instance parameter. At node $(j, 2)$, the vessel starts to operate (represented by o), and then takes the waiting arc, reaching node $(j, 3)$, where no operation takes place. After, it departs to port i, taking $T_{jiv} = 2$ time periods to travel between the ports. At port i in time $t = 5$, vessel v operates and finally ends its route departing to sink node $d(v)$. Note that a vessel can operate in a port and depart to another port in the same time period. The notation for the time-space model is presented in table 3.1.

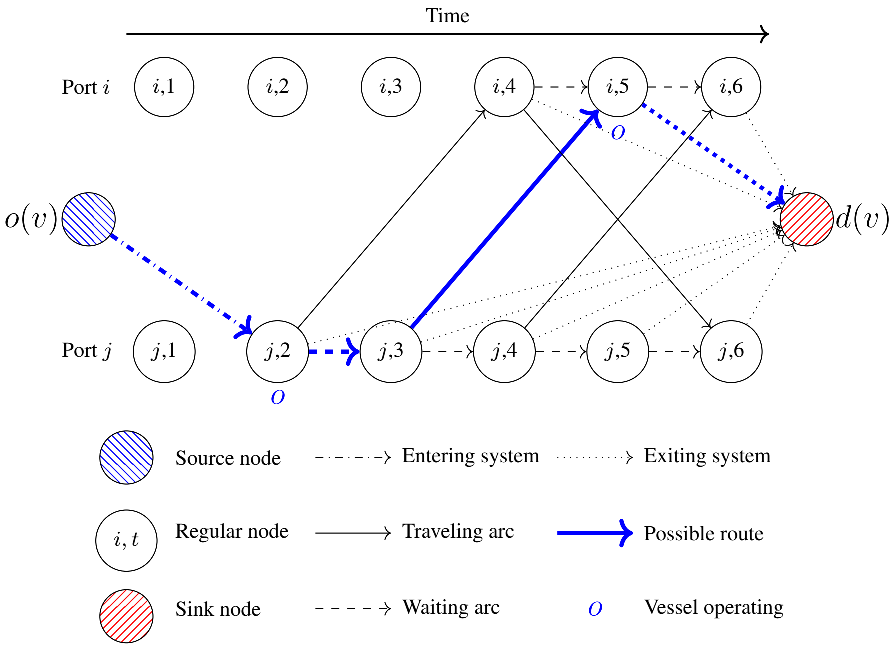
In the TS model presented in Papageorgiou et al. (2014b), the minimum (maximum) amount of product operated are associated with a time index t, i.e., $F_{it}^{\min}(F_{it}^{\max})$. We omit the time index as the instances proposed by Papageorgiou et al. (2014b) consider these parameters as constant values along the planning horizon.

Table 3.1: Time-space network model notation.

Index and sets	
$t \in \mathcal{T}$	set of time periods with T = $\|\mathcal{T}\|$;
$v \in \mathcal{V}$	set of vessels;
$i \in \mathcal{J}^{\mathrm{P}}$	set of production or loading ports;
$i \in \mathcal{J}^{\mathrm{C}}$	set of consumption or discharging ports;
$i \in \mathcal{J}$	set of all ports: $\mathcal{J} = \mathcal{J}^{\mathrm{P}} \cup \mathcal{J}^{\mathrm{C}}$;
$o(v)$	dummy source node for vessel v;
$d(v)$	dummy sink node for vessel v;
Parameters	
J_v^0	initial port of vessel v;
T_v^0	first time avaiable of vessel v;
B_i	number of berths (berth limit) at port i;
T_{ijv}	traveling time between port i and j for vessel v;
C_{ijv}	traveling cost between port i and j for vessel v;
D_{it}	production (consumption) rate at port i in period t;
Δ_i	+1 if $i \in \mathcal{J}^{\mathrm{P}}$ and -1 if $i \in \mathcal{J}^{\mathrm{C}}$;
$F_i^{\min}(F_i^{\max})$	minimum (maximum) amount that can be loaded or discharged at port i from a single vessel in each time period;
Q_v	capacity of vessel v;
R_{it}	the unit sales revenue for product discharged at port i in time period t;
$S_{it}^{\min}(S_{it}^{\max})$	lower bound (capacity) at port i at time t;
S_i^0	initial inventory at port i;
L_v^0	initial load on board vessel v;
P_{it}	penalization per unit of inventory violation at port i in time period t;
$\alpha_{it}^{\max}$	maximum violation of the inventory at port i in time period t;
$\alpha_i^{\max}$	maximum violation of the inventory at port i in the entire planning horizon;
ϵ	nonnegative cost associated with attempting to load or discharge at a port;
Decision variables	
f_{it}^v	(continuous) amount loaded/discharged by vessel v in port i at period t;
s_{it}	(continuous) inventory at port i available at the end of period t;
s_t^v	(continuous) load on board vessel v available at end of period t;
α_{it}	(continuous) amount bought from of sold to a spot market by port i on time t;
x_{ijvt}	(binary) 1 if vessel v travels from port i to port j departing at time t, 0 otherwise;
w_{ivt}	(binary) 1 if vessel v waits outside port i in time period t, 0 otherwise;
o_{ivt}	(binary) 1 if vessel v operates at port i in time period t, 0 otherwise;

Source: From the author (2020).

Figure 3.1: Example of a TS model for MIRP with two ports and one vessel.



Source: From the author (2020).

The TS model for the MIRP can be formulated as follows:

$$\max \begin{aligned} &\sum_{i\in\mathcal{J}^{\mathrm{C}}}\sum_{t\in\mathcal{T}}\sum_{v\in\mathcal{V}} R_{it} f_{it}^{v} - \sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{J}\cup\{o(v)\}}\sum_{j\in\mathcal{J}\cup\{d(v)\}}\sum_{t\in\mathcal{T}} C_{ijv} x_{ijvt} \\ &- \sum_{i\in\mathcal{J}}\sum_{t\in\mathcal{T}}\sum_{v\in\mathcal{V}} (t\epsilon) o_{ivt} - \sum_{i\in\mathcal{J}}\sum_{t\in\mathcal{T}} P_{it}\alpha_{it} \end{aligned} \tag{3.1}$$

$$\text{s.t. } x_{o(v)J_v^0 v0} = 1, \ \ \forall v \in \mathcal{V} \tag{3.2}$$

$$\begin{aligned} &\sum_{j\in\mathcal{J}\cup\{o(v)\}} x_{jiv,t-T_{jiv}} + w_{iv,t-1} = \\ &\sum_{j\in\mathcal{J}\cup\{d(v)\}} x_{ijvt} + w_{ivt}, \ \ \forall i \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} : t \geq T_v^0 \end{aligned} \tag{3.3}$$

$$\sum_{i\in\mathcal{J}}\sum_{t\in\mathcal{T}} x_{id(v)vt} = 1, \ \ \forall v \in \mathcal{V} \tag{3.4}$$

$$s_{it} = s_{i,t-1} + \Delta_i \left(D_{it} - \sum_{v\in\mathcal{V}} f_{it}^{v} - \alpha_{it} \right), \ \ \forall\, i \in \mathcal{J}, t \in \mathcal{T} \tag{3.5}$$

$$s_{i,0} = S_i^0, \ \ \forall i \in \mathcal{J} \tag{3.6}$$

$$s_t^v = s_{t-1}^v + \sum_{i\in\mathcal{J}} \Delta_i f_{it}^v, \ \ \forall\, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.7}$$

$$s_1^v = L_v^0, \ \ \forall v \in \mathcal{V} \tag{3.8}$$

$$\sum_{v\in\mathcal{V}} o_{ivt} \leq B_i \ \ \forall\, i \in \mathcal{J}, t \in \mathcal{T} \tag{3.9}$$

$$o_{ivt} \leq \sum_{j\in\mathcal{J}} x_{ijvt}, \ \ \forall\, i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.10}$$

$$s_t^v \geq Q^v x_{ijvt}, \ \ \forall\, v \in \mathcal{V}, i \in \mathcal{J}^{\mathrm{P}}, j \in \mathcal{J}^{\mathrm{C}} \cup \{d(v)\}, t \in \mathcal{T} \tag{3.11}$$

$$s_t^v \leq Q^v(1 - x_{ijvt}), \ \ \forall\, v \in \mathcal{V}, i \in \mathcal{J}^{\mathrm{C}}, j \in \mathcal{J}^{\mathrm{P}} \cup \{d(v)\}, t \in \mathcal{T} \tag{3.12}$$

$$\sum_{t\in\mathcal{T}} \alpha_{it} \leq \alpha_i^{\max}, \ \ \forall\, i \in \mathcal{J} \tag{3.13}$$

$$0 \leq \alpha_{it} \leq \alpha_{it}^{\max}, \ \ \forall\, i \in \mathcal{J}, t \in \mathcal{T} \tag{3.14}$$

$$F_i^{\min} o_{ivt} \leq f_{it}^v \leq F_i^{\max} o_{ivt}, \ \ \forall\, i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.15}$$

$$S_{it}^{\min} \leq s_{it} \leq S_{it}^{\max}, \ \ \forall\, i \in \mathcal{J}, t \in \mathcal{T} \tag{3.16}$$

$$0 \leq s_t^v \leq Q^v, \ \ \forall\, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.17}$$

$$x_{ijvt} \in \{0,1\}, \ \ \forall\, i \in \mathcal{J} \cup \{o(v)\}, j \in \mathcal{J} \cup \{d(v)\}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.18}$$

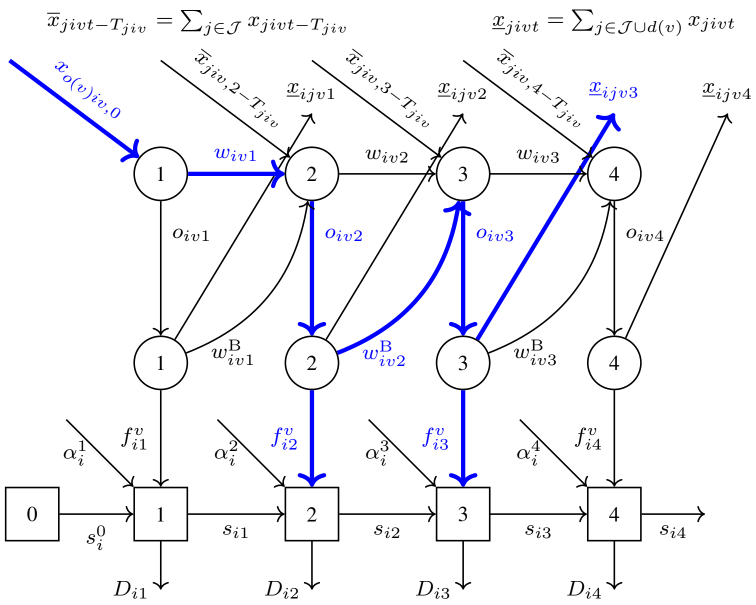
$$o_{ivt}, w_{ivt} \in \{0,1\}, \ \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.19}$$

Objective function (3.1) maximizes the revenue obtained by unloading product at discharging ports (first term), minus the traveling costs (second term), and the penalization costs (third and fourth term). The third term is a negligible value accounted for when an operation takes place. It induces vessels to operate as soon and few times as possible, The fourth term is the penalization by using spot markets, i.e., violating port-time inventory constraints. Constraints (3.2) fix the initial port-time node of each vessel, departing from the source node $o(v)$. The departing time from the source node is $t = 0$, and the traveling time between $o(v)$ and initial port J_v^0 is the first time in which vessel v is available (T_v^0). Constraints (3.3) are the vessels' flow balance at the regular nodes, which consider the traveling arcs and the waiting arc that enter and leave each node. Constraint (3.4) is the flow balance at the sink node. Constraints (3.5) impose the port's inventory balance at the end of each time period. Constraints (3.6) and (3.8) define the initial inventory at each port and the load on board each vessel, respectively. Constraints (3.7) define the vessel load balance at the end of each time period. Constraints (3.9) impose the maximum number of vessels that can operate simultaneously in each port. Constraints (3.10) require that a vessel can only operate at a node if it is actually at that node. Constraints (3.11) define that vessels must travel at the maximum capacity if sailing from a loading port to a discharging port or the sink node, while constraints (3.12) require that vessels must be empty if traveling from a discharging port to a loading port or the sink node. Constraints (3.13) limit a port's cumulative amount sold to or bought from a spot market in the entire planning horizon, and constraints (3.14) limit the amount in each time period. Constraints (3.15) impose that once a vessel operates at a port, the operated amount must lie between upper and lower limits. Constraints (3.16) and (3.17) assure that inventories at ports and vessel loads are kept between their lower and upper limits in each time period, respectively. Equations (3.18) and (3.19) ensure the integrality constraints on the binary variables.

3.2 Fixed-Charge Network Flow Model

Agra et al. (2013) proposed a single-commodity fixed-charge network flow model (FCNF) to derive a formulation with better linear relaxation than the corresponding time-space network, considering another MIRP variation. In the FCNF, a commodity supplied by loading ports flows along the arcs corresponding to the vessel routes until reaching discharging ports to be consumed externally. Figure 3.2 illustrates the FCNF model structure for a discharging port i and a vessel v, highlighting a possible solution.

Figure 3.2: Example of a FCNF model for a discharging port i and vessel v.



Source:From the author (2020).

In Figure 3.2 each port-time $(i, t), i \in \mathcal{J}, t \in \mathcal{T}$ is divided into three levels. The top-level coordinates the vessels' arrival at the port, while the middle level coordinates the port's operations and departures. The bottom layer is responsible for the inventory at the port. The highlighted arcs represent a possible flow solution. When vessel v arrives at specific node (i, t), it can wait for one time-period ($w_{ivt} = 1$), or it can operate ($o_{ivt} = 1$), where the commodity flows to the middle level. At the middle level, the flow can be divided between the discharging amount f_{it}^v, which flows to the bottom level, and the remaining amount, which represents the load still on board the vessel. Then, vessel v can wait one time period (using another set of waitng variables $w_{ivt}^{\mathrm{B}} = 1$), or can depart to another port/finish its route through arc $x_{ijvt}, j \in \mathcal{J} \cup d(v)$. Variables α_{it} flow directly to the bottom level as they work as slack variables.

The FCNF structure accommodates practical aspects of the real problem:

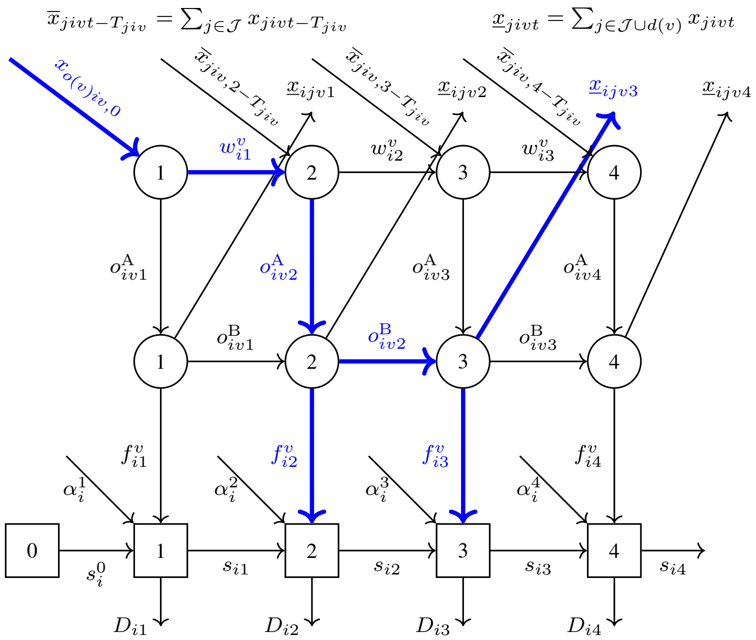
Assumption 1 *a vessel cannot visit a port without operating at this port, since vessel voyages are quite expensive;*

Assumption 2 *when a vessel ends its operation at the port, it must depart to another port or end its route.*

Besides assumptions 1 and 2 not being constraints in model (3.1)-(3.19), they do not cut a possible optimal solution. The traveling arcs' high cost suggests that a vessel will not travel to a port if it does not operate at such a port. It just increases the objective function's cost, considering that the triangular inequality is respected. Assumption 2 reduces the possibility of symmetric solutions. For example, after operated at port i, the vessel travels to port j and waits l times before operating at j. Without Assumption 2, the vessel can wait at port i for l times after operated and then departs to port j, operating at j in the same time of the arrival, which is a symmetric solution for this MIRP.

It is possible to consider a more restrictive practical assumption on the FCNF model by modifying the structure described in Figure 3.2. Such an assumption imposes that once a vessel started to operate at a port, it cannot wait one or more time periods and then operate again, i.e., if a vessel operates in two or more time periods at the same port visit, such operations must be consecutive. The assumption is illustrated in Figure 3.3.

Figure 3.3: Example of a FCNF+ model for a discharging port i and vessel v.



Source: From the author (2020).

In the FCNF model ilustrated in Figure3.3, variable w^{B}_{ivt} is replaced by variable o^{B}_{ivt} which indicates if vessel v continues to operate at port i after started to operate in a time period before t, i.e. $o^{\mathrm{A}}_{ivt} = 1$.

Along with this thesis, we refer to the model in which structure is illustrated in Figure 3.2 as just **FCNF**, while we refer to **FCNF+** as the model in which the structure is illustrated in Figure 3.3 to represent that it is more restrictive than the FCNF.

In the FCNF formulation the flow conservation constraints (3.3) are replaced by the following constraints:

$$\sum_{j \in \mathcal{J} \cup \{o(v)\}} x_{jiv,t-T_{jiv}} + w_{iv,t-1} = w_{ivt} + o_{ivt}, \ \ \forall i \in \mathcal{J}, v \in \mathcal{V}, t \in \mathcal{T} : t \geq T_v^0 \quad (3.20)$$

$$o_{ivt} = w_{ivt}^{\mathrm{B}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} x_{ijvt}, \ \ \forall i \in \mathcal{J} \cup \{d(v)\}, t \in \mathcal{T}, v \in \mathcal{V} \quad (3.21)$$

$$w_{ivt}^{\mathrm{B}} \in \{0, 1\}, \ \ \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V}. \quad (3.22)$$

Constraints (3.20) and (3.21) are the flow balance at the top and middle level, respectively. Constraints (3.22) define the waiting after operation variables' scope. The following constraints represent the corresponding constraints for the FCNF+ model:

$$\sum_{j \in \mathcal{J} \cup \{o(v)\}} x_{jiv,t-T_{jiv}} + w_{iv,t-1} = w_{ivt} + o_{ivt}^{\mathrm{A}}, \ \ \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} : t \geq T_v^0 \quad (3.23)$$

$$o_{ivt}^{\mathrm{A}} + o_{iv,t-1}^{\mathrm{B}} = o_{ivt}^{\mathrm{B}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} x_{ijvt}, \ \ \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \quad (3.24)$$

$$o_{ivt}^{\mathrm{A}} + o_{iv,t-1}^{\mathrm{B}} = o_{ivt}, \ \ \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \quad (3.25)$$

$$o_{ivt}^{\mathrm{A}}, o_{ivt}^{\mathrm{B}} \in \{0, 1\}, \ \ \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V}. \quad (3.26)$$

Constraints (3.25) defines that if a vessel is operating, it started to operate in the current time period, or it continues to operate from the previous time period.

Besides the additional binary variables, new continuous flow variables are associated with such variables. For the FCNF and FCNF+ models:

- f_{ijvt}^{X} (associated with x_{ijvt}) is the load on board at vessel v when it travels from port i to port j, starting at time t.

- f_{ivt}^{O} (associated with o_{ivt}) corresponds to the load on board at vessel v when it operates at port i in time period t.

- f_{ivt}^{W} (associated with w_{ivt}) is the load on board at vessel v while waiting during time period t at port i.

For the FCNF model:

- f_{ivt}^{WB} (associated with w_{ivt}^{B}) represents the load on board at vessel after it operated at time $t-1$ and continues at the same port.

For the FCNF+ model:

- f_{ivt}^{OA} (associated with variable o_{ivt}^{A}.) corresponds to the load on board at vessel v when it starts to operate at port i in time period t and it has not operated at time $t-1$.

- f_{ivt}^{OB} (associated with variable o_{ivt}^{W}) represents the load on board at vessel after it operated at time $t-1$, and before it continues operating at time t.

As the model considers the commodity flow, the vessel inventory variable s_t^v is removed and the commodity flow conservation is modeled as follows for the FCNF model:

$$\sum_{j \in \mathcal{J} \cup \{o(v)\}} f_{jiv,t-T_{jiv}}^{\mathrm{X}} + f_{iv,t-1}^{\mathrm{W}} = f_{ivt}^{\mathrm{W}} + f_{ivt}^{\mathrm{O}}, \quad \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.27}$$

$$f_{ivt}^{\mathrm{O}} + \Delta_i f_{it}^v = f_{ivt}^{\mathrm{WB}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} f_{ijvt}^{\mathrm{X}}, \quad \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.28}$$

Constraints (3.26) and (3.27) define the flow conservation of the commodity on board each vessel along the nodes. For the FCNF+ model, the corresponding constraints are:

$$\sum_{j \in \mathcal{J} \cup \{o(v)\}} f_{jiv,t-T_{jiv}}^{\mathrm{X}} + f_{iv,t-1}^{\mathrm{W}} = f_{ivt}^{\mathrm{W}} + f_{ivt}^{\mathrm{OA}}, \quad \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.29}$$

$$f_{ivt}^{\mathrm{OA}} + f_{iv,t-1}^{\mathrm{OB}} + \Delta_i f_{it}^v = f_{ivt}^{\mathrm{OB}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} f_{ijvt}^{\mathrm{X}}, \quad \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.30}$$

$$\tag{3.31}$$

Constraints (3.31) define the initial load on board each vessel for both models.

$$f_{o(v)jvt}^{X} = L_v^0 x_{o(v)jvt}, \quad \forall j \in \mathcal{J} \cup \{d(v)\}, t \in \mathcal{T}, v \in \mathcal{V} \tag{3.32}$$

The following constraints bound the variables of FCNF and FCNF+ models:

$$0 \leq f_{ijvt}^{X} \leq Q_v x_{ijvt}, \forall v \in \mathcal{V}, i \in \mathcal{J} \cup \{o(v)\}, j \in \mathcal{J} \cup \{d(v)\}, t \in \mathcal{T}, \tag{3.33}$$

$$0 \leq f_{ivt}^{\mathrm{W}} \leq Q_v w_{ivt}, \forall v \in \mathcal{V}, i \in \mathcal{J}, t \in \mathcal{T}. \tag{3.34}$$

Constraints (3.34)-(3.35) bounds the specific variables of the FCNF model:

$$0 \leq f_{ivt}^{\mathrm{O}} \leq Q_v o_{ivt}, \forall v \in \mathcal{V}, i \in \mathcal{J}, t \in \mathcal{T}, \tag{3.35}$$

$$0 \leq f_{ivt}^{\mathrm{WB}} \leq Q_v w_{ivt}^{\mathrm{B}}, \forall v \in \mathcal{V}, i \in \mathcal{J}, t \in \mathcal{T} \tag{3.36}$$

And the following constraints bounds the specific variables of the FCNF+ model:

$$0 \leq f_{ivt}^{\mathrm{OA}} \leq Q^v o_{ivt}^{\mathrm{A}}, \forall v \in \mathcal{V}, i \in \mathcal{J}, t \in \mathcal{T}, \quad (3.37)$$

$$0 \leq f_{ivt}^{\mathrm{OB}} \leq Q^v o_{ivt}^{\mathrm{B}}, \forall v \in \mathcal{V}, i \in \mathcal{J}, t \in \mathcal{T} \quad (3.38)$$

Finally, constraints (3.11) and (3.12) are reformulated as follows:

$$f_{ijvt}^{\mathrm{X}} \geq Q^v x_{ijvt}, \ \ \forall v \in \mathcal{V}, t \in \mathcal{T}, i \in \mathcal{J}^P, j \in \mathcal{J}^D \cup \{d(v)\}, \quad (3.39)$$

$$f_{ijvt}^{\mathrm{X}} \leq Q^v (1 - x_{ijvt}), \ \ \forall v \in \mathcal{V}, t \in \mathcal{T}, i \in \mathcal{J}^D, j \in \mathcal{J}^P \cup \{d(v)\}. \quad (3.40)$$

The FCNF formulation is refereed by equations (3.1)–(3.4), (3.5)–(3.6), (3.9), (3.13)–(3.16), (3.18)–(3.22), (3.26)–(3.27), (3.31)–(3.35), (3.39)–(3.40). The FCNF+ formulation is refereed by equations (3.1)–(3.4), (3.5)–(3.6), (3.9), (3.13)–(3.16), (3.18), (3.23)–(3.26), (3.29)–(3.30), (3.31)–(3.33), (3.37)–(3.40).

3.2.1 Tightening Flow Variables Bounds

We can tighten the upper and lower limits of the flow variables based on the assumptions 1 and 2, to also tighten the linear relaxation value and possibly improving our algorithms' performance. We present the tightening constraints for the FCNF model. They were not considered in the FCNF+ model. We use the limits on flow variables f_{ivt}^{WB} that represents the load on board at vessel v after it operated at port i at time $t-1$ to explain the idea. Knowing that f_{ivt}^{WB} must be positive if vessel v waits after operating at port i, at least the value $F_i^{\min}$ will be operated before and after waiting. Therefore, we can affirm that $f_{ivt}^{\mathrm{WB}} \geq F_i^{\min}$ if i is a loading port, and $f_{ivt}^{\mathrm{WB}} \leq Q_v - F_i^{\min}$ if i is a discharging port. Consequently, vessel v must have a maximum load $Q_v - F_i^{\min}$ to be allowed to load again at $i \in \mathcal{J}^{\mathrm{P}}$, or must have a minimum load $F_i^{\min}$ to be allowed to discharge again at the port $i \in \mathcal{J}^{\mathrm{C}}$. Thus, flow variable f_{ivt}^{WB} can be bounded as follows:

$$w_{ivt}^{\mathrm{B}} F_i^{\min} \leq f_{ivt}^{\mathrm{WB}} \leq w_{ivt}^{\mathrm{B}} (Q_v - F_i^{\min}), \ \ \forall v \in \mathcal{V}, t \in \mathcal{T}, i \in \mathcal{J}. \quad (3.41)$$

For the other flow variables, the limits are different depending on the type of port. For the loading ports, the new limits are defined as follows:

$$x_{ijvt}F_i^{\min} \leq f_{ijvt}^{\mathrm{X}} \leq x_{ijvt}(Q_v - F_j^{\min}), \;\; \forall v \in \mathcal{V}, t \in \mathcal{T}, i,j \in \mathcal{J}^P, \tag{3.42}$$

$$0 \leq f_{ivt}^{\mathrm{W}} \leq w_{vit}(Q_v - F_i^{\min}), \;\; \forall v \in \mathcal{V}, t \in \mathcal{T}, i \in \mathcal{J}^P, \tag{3.43}$$

$$0 \leq f_{ivt}^{\mathrm{O}} \leq o_{ivt}(Q_v - F_i^{\min}), \;\; \forall v \in \mathcal{V}, t \in \mathcal{T}, i \in \mathcal{J}^P. \tag{3.44}$$

Considering the discharging ports, the new limits are:

$$x_{ijvt}F_j^{\min} \leq f_{ijvt}^{\mathrm{X}} \leq x_{ijvt}(Q_v - F_i^{\min}), \;\; \forall v \in \mathcal{V}, t \in \mathcal{T}, i,j \in \mathcal{J}^D, \tag{3.45}$$

$$w_{ivt}F_i^{\min} \leq f_{ivt}^{\mathrm{W}} \leq w_{vit}Q_v, \;\; \forall v \in \mathcal{V}, t \in \mathcal{T}, i \in \mathcal{J}^D, \tag{3.46}$$

$$o_{ivt}F_i^{\min} \leq f_{ivt}^{\mathrm{O}} \leq o_{ivt}Q_v, \;\; \forall v \in \mathcal{V}, t \in \mathcal{T}, i \in \mathcal{J}^D. \tag{3.47}$$

The use of lower limits different from zero in the flow variables implies adding $O(|\mathcal{V}||\mathcal{T}||\mathcal{J}|^2)$ new constraints to the FCNF model.

3.3 Models Size

The number of variables and constraints of both FCNF and TS models in order terms is $O(|\mathcal{V}||\mathcal{T}||\mathcal{J}|^2)$. However, the FCNF model has more variables and constraints than the TS model, as a new set of wait variables w_{ivt}^{B} is created. Also, for each binary variable (except o_{ivt}) is associated a linear variable for modeling the flow of vessel, thus replacing the $O(|\mathcal{V}||\mathcal{T}|)$ vessel invetory variables s_t^v by $O(|\mathcal{V}||\mathcal{T}||\mathcal{J}|^2)$ flow variables. For each linear flow variable, there is a new constraint for bounding it. Consequently, the number of constraints in the FCNF model also grows compared to the time-space model. Compared to the FCNF model, the FCNF+ model has additionally $O(|\mathcal{V}||\mathcal{J}||\mathcal{T}|)$ variables, and $O(|\mathcal{V}||\mathcal{J}||\mathcal{T}|)$ constraints (due the constraints (3.25)).

3.4 Valid Inequalities

Valid inequalities are used for tightening the FCNF and TS formulations. Several works proposed valid inequalities for different variations of the MIRP as observed in Chapter 2. In this section, valid inequalities based on knapsack sets derived from the work of Agra et al. (2013) are presented. We present their construction based on the FCNF+ model and adapt them for the FCNF and TS models.

3.4.1 Loading Ports

For each loading port $i \in \mathcal{J}^{\mathrm{P}}$, the valid inequalities impose a minimum flow out of the ports considering a specific time interval. Let $\mathcal{T}' = [l, k] \subseteq \mathcal{T}$ be such time interval. Also, let $\mathcal{T}'_v \subseteq \mathcal{T}'$ be a subset of the time periods in $\mathcal{T}'$ in which vessel $v \in \mathcal{V}$ is assumed to operate at port i. Additionaly, consider $\mathcal{T}'^{+}_v = \{t \in \mathcal{T}'_v : t+1 \notin \mathcal{T}'_v\}$ as the time periods in $\mathcal{T}'$ followed by the departure of vessel v from port i. Set $\mathcal{T}'^{-}_v = \{t \in \mathcal{T}'_v : t-1 \notin \mathcal{T}'_v\}$ represents the time periods in $\mathcal{T}'$ in which vessel v starts to load at i. Summing the flow conservation constraints (3.27) for loading port i over all vessels and time periods $t-1 \in \mathcal{T}_v$ we obtain

$$\begin{aligned}\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'_v} f^v_{it} = &\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'_v}(f^{\mathrm{OB}}_{ivt} - f^{\mathrm{OB}}_{iv,t-1}) \\ &+ \sum_{v\in\mathcal{V}}\sum_{j\in\mathcal{J}\cup\{d(v)\}}\sum_{t\in\mathcal{T}'_v} f^X_{ijvt} - \sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'_v} f^{\mathrm{OA}}_{ivt}.\end{aligned}$$

Considering

$$\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'_v}(f^{\mathrm{OB}}_{ivt} - f^{\mathrm{OB}}_{iv,t-1}) = \sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'^{+}_v} f^{\mathrm{OB}}_{ivt} - \sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'^{-}_v} f^{\mathrm{OB}}_{iv,t-1}$$

and assuming that f^{OA}_{ivt} and $f^{\mathrm{OB}}_{iv,t-1}$ are nonnegative gives

$$\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'} f^v_{it} \le \sum_{v\in\mathcal{V}}\left(\sum_{t\in\mathcal{T}'^{+}_v} f^{\mathrm{OB}}_{ivt} + \sum_{j\in\mathcal{J}\cup\{d(v)\}}\sum_{t\in\mathcal{T}'_v} f^X_{ijvt} + \sum_{t\in\mathcal{T}'\setminus\mathcal{T}'_v} f^v_{it}\right) \tag{3.48}$$

Summing the inventory constraints (3.5) over $\mathcal{T}'$ gives

$$\sum_{v\in\mathcal{V}}\sum_{t\in\mathcal{T}'} f^v_{it} = \sum_{t\in\mathcal{T}'}(-s_{it} + s_{i,t-1} + D_{ij} - \alpha_{it}) \tag{3.49}$$

Using $S^{\min}_{it}$ as an underestimator of s_{it} and and $\overline{\alpha}^{\max}_i = \min\{\alpha^{\max}_i, \sum_{t\in\mathcal{T}'}\alpha^{\max}_{it}\}$ as the maximum amount that can be bought from or sold to a spot market by port i in time interval $\mathcal{T}'$, the combination with (3.48) and (3.49) results in

$$s_{ik} + \sum_{v \in \mathcal{V}} \left(\sum_{t \in \mathcal{T'}_v^+} f_{ivt}^{\mathrm{OB}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} \sum_{t \in \mathcal{T'}_v} f_{ijvt}^{X} + \sum_{t \in \mathcal{T}' \setminus \mathcal{T'}_v} f_{it}^{v} \right) \geq \sum_{i \in \mathcal{T}'} D_{it} + S_{i,l-1}^{\min} - \overline{\alpha}_i^{\max} \tag{3.50}$$

Using upper bounds on flow variables given by constraints (3.15) and (3.32)-(3.33), constraint (3.50) becomes in:

$$\begin{aligned} s_{ik} + \sum_{v \in \mathcal{V}} \left(\sum_{t \in \mathcal{T'}_v^+} Q_v o_{ivt}^{\mathrm{B}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} \sum_{t \in \mathcal{T'}_v} Q_v x_{ijvt} + \sum_{t \in \mathcal{T}' \setminus \mathcal{T'}_v} F_{it}^{\max} o_{ivt} \right) \\ \geq \sum_{t \in \mathcal{T}'} D_{it} + S_{i,l-1}^{\min} - \overline{\alpha}_i^{\max} \end{aligned} \tag{3.51}$$

Replacing variable s_{ik} by its upper bound $S_{ik}^{\max}$ gives knapsack sets valid for the set of feasible solutions of FCNF+. For an arbitrary $Q > 0$, we can construct the following Chvatal-Gomory inequalities:

$$\begin{aligned} \sum_{v \in \mathcal{V}} \Bigg(\sum_{t \in \mathcal{T'}_v^+} \left\lceil \frac{Q_v}{Q} \right\rceil o_{ivt}^{\mathrm{B}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} \sum_{t \in \mathcal{T'}_v} \left\lceil \frac{Q_v}{Q} \right\rceil x_{ijvt} \\ + \sum_{t \in \mathcal{T}' \setminus \mathcal{T'}_v} \left\lceil \frac{F_i^{\max}}{Q} \right\rceil o_{ivt} \Bigg) \geq \left\lceil \frac{\sum_{t \in \mathcal{T}'} D_{ij} + S_{i,l-1}^{\min} - S_{ik}^{\max} - \overline{\alpha}_i^{\max}}{Q} \right\rceil \end{aligned} \tag{3.52}$$

There are two cases of inequality (3.52) that lead to more simple inequalities. For the first case, consider that $\mathcal{T'}_v = \mathcal{T}'$ implies in $\mathcal{T'}_v^+ = k$, and $\mathcal{T}' \setminus \mathcal{T'}_v = \emptyset$. Defining $\overline{Q} = \max\{Q_v : v \in V\}$ as the maximum vessel capacity gives:

$$\sum_{v \in \mathcal{V}} \left(o_{ivk}^{\mathrm{B}} + \sum_{t \in \mathcal{T}'} \sum_{j \in \mathcal{J} \cup \{d(v)\}} x_{ijvt} \right) \geq \left\lceil \frac{\sum_{t \in \mathcal{T}'} D_{it} + S_i^{\min} - S_i^{\max} - \overline{\alpha}_i^{\max}}{\overline{Q}} \right\rceil \tag{3.53}$$

The second case considers $\mathcal{T'}_v = \emptyset$, implying in $\mathcal{T'}_v^+ = \emptyset$ and $\mathcal{T}' \setminus \mathcal{T'}_v = \mathcal{T}'$. The corresponding knapsack inequality is:

$$\sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}'} o_{ivt} \geq \left\lceil \frac{\sum_{t \in \mathcal{T}'} D_{it} + S_i^{\min} - S_i^{\max} - \overline{\alpha}_i^{\max}}{F_i^{\max}} \right\rceil \tag{3.54}$$

Inequalities (3.53) impose a minimum number of vessel departures and operations in the time interval $\mathcal{T}'$, while (3.54) impose a minimum number of operations (loading) at port i.

3.4.2 Discharging Ports

In discharging ports, the valid inequalities should consider the subgraph entering arcs, using the three levels of FCNF+ formulation, including the entering traveling arcs and waiting arcs. Consider again the subset of time periods $\mathcal{T}' = [l, k]$.

The following disjoint sets can be defined in the subgraph for all $t \in \mathcal{T}'$: R_v^0, R_v^1, and R_v^2. If $t \in R_v^0$, only the bottom level node of time t is included in the subgraph. If $t \in R_v^1$, the bottom and middle-level nodes of t are included in the subgraph. If $t \in R_v^2$, the bottom, middle, and top-level nodes of t are included in the subgraph. Finally, define ${\mathcal{T}'}_v^1 = \{t \in R_v^1 \cup R_v^2 : t-1 \notin (R_v^1 \cup R_v^2)\}$, and ${\mathcal{T}'}_v^2 = \{t \in R_v^2 : t-1 \notin R_v^2\}$.

Summing over constraints (3.5) for discharging ports ($\Delta = -1$), and assuming $S_{ik}^{\min}$ as the lower bound of s_{ik} gives

$$s_{i,l-1} + \sum_{t \in \mathcal{T}'} \left(\sum_{v \in \mathcal{V}} f_{it}^v + \alpha_{it} \right) \geq \sum_{t \in \mathcal{T}'} D_{it} + S_{ik}^{\min} \tag{3.55}$$

writing in terms of the partition R_v^0, R_v^1, R_v^2 gives

$$s_{i,l-1} + \sum_{v \in \mathcal{V}} \left(\sum_{t \in R_v^0} f_{it}^v + \sum_{t \in R_v^1} f_{it}^v + \sum_{t \in R_v^2} f_{it}^v \right) + \sum_{t \in \mathcal{T}'} \alpha_{it} \geq \sum_{t \in \mathcal{T}'} D_{it} + S_{ik}^{\min} \tag{3.56}$$

Summing constraints (3.26) and (3.27) over R_v^2 and $R_v^2 \cup R_v^1$ gives, respectively:

$$\sum_{t \in R_v^2} f_{ivt}^{\mathrm{OA}} = \sum_{t \in R_v^2} \left(\sum_{j \in \mathcal{J} \cup \{o(v)\}} f_{jiv,t-T_{jiv}}^X + f_{iv,t-1}^{\mathrm{W}} - f_{ivt}^{\mathrm{W}} \right) \tag{3.57}$$

$$\sum_{t \in R_v^2 \cup R_v^1} \left(f_{ivt}^{\mathrm{OA}} + f_{iv,t-1}^{\mathrm{OB}} - f_{it}^v \right) = \sum_{t \in R_v^2 \cup R_v^1} \left(f_{ivt}^{\mathrm{OB}} + \sum_{j \in \mathcal{J} \cup \{d(v)\}} f_{ijvt}^X \right) \tag{3.58}$$

We can cancel f_{ivt}^{W} (just considering the flow o_{ivt}^A that will enter in a port) to simplifly equation (3.57), while equation (3.58) is simplified by canceling f_{ivt}^{OB}. Using the nonnegativity of variables $f_{ivt}^{W}, f_{ijvt}^{X}$ and f_{ivt}^{OB} one can obtain from equation (3.56):

$$\begin{aligned} s_{i,l-1} + \sum_{v \in \mathcal{V}} \Bigg(\sum_{t \in R_v^0} f_{it}^{v} + \sum_{t \in R_v^1} f_{ivt}^{\mathrm{OA}} + \sum_{t \in R_v^2} \sum_{j \in \mathcal{J} \cup \{o(v)\}} f_{jiv,t-T_{jiv}}^{X} \\ + \sum_{t \in \mathcal{T'}_v^2} \left(f_{iv,t-1}^{\mathrm{W}} \right) + \sum_{t \in \mathcal{T'}_v^1} f_{iv,t-1}^{\mathrm{OB}} \Bigg) + \sum_{t \in \mathcal{T}'} \alpha_{it} \geq \sum_{t \in \mathcal{T}'} D_{it} + S_{ik}^{\min} \end{aligned} \tag{3.59}$$

Using the variable upper bound (3.15) and (3.32)-(3.33), and defining $\overline{\alpha}_i^{\max} = \min\{\alpha_i^{\max}, \sum_{t \in \mathcal{T}'} \alpha_{it}^{\max}\}$ as the maximum amount that can be bought from or sold to a spot market by port i in time interval $\mathcal{T}'$, inequality (3.59) becomes:

$$\begin{aligned} s_{i,l-1} + \sum_{v \in \mathcal{V}} \Bigg(\sum_{t \in R_v^0} F_i^{\max} o_{ivt} + \sum_{t \in R_v^1} Q_v o_{ivt}^{A} + \sum_{t \in R_v^2} \sum_{j \in \mathcal{J} \cup \{o(v)\}} Q_v x_{jiv,t-T_{jiv}} \\ + \sum_{t \in \mathcal{T'}_v^2} (Q_v w_{iv,t-1}) + \sum_{t \in \mathcal{T'}_v^1} Q_v o_{iv,t-1}^{B} \Bigg) \geq \sum_{t \in \mathcal{T}'} D_{it} + S_{ik}^{\min} - \overline{\alpha}_i^{\max} \end{aligned} \tag{3.60}$$

Replacing $s_{i,l-1}$ by its upper bound $S_{i,l-1}^{\max}$ if $l > 1$, and by S_i^0 if $l = 1$), integer knapsack constraints are obtained. For an arbitrary $Q > 0$, the following Chvatal-Gomory inequalities are obtained:

$$\begin{aligned} \sum_{v \in \mathcal{V}} \Bigg(\sum_{t \in R_v^0} \left\lceil \frac{F_i^{\max}}{Q} o_{ivt} \right\rceil + \sum_{t \in R_v^1} \left\lceil \frac{Q_v}{Q} \right\rceil o_{ivt}^{A} + \sum_{t \in R_v^2} \sum_{j \in \mathcal{J} \cup \{o(v)\}} \left\lceil \frac{Q_v}{Q} \right\rceil x_{jiv,t-T_{ijv}} \\ + \sum_{t \in \mathcal{T'}_v^2} \left\lceil \frac{Q_v}{Q} \right\rceil w_{iv,t-1} + \sum_{t \in \mathcal{T'}_v^1} \left\lceil \frac{Q_v}{Q} \right\rceil o_{iv,t-1}^{B} \Bigg) \geq \left\lceil \frac{\sum_{t \in \mathcal{T}'} D_{it} - S_{i,l-1}^{\max} + S_{ik}^{\min} - \overline{\alpha}_i^{\max}}{Q} \right\rceil \end{aligned} \tag{3.61}$$

Three special cases can be derived from (3.61): Considering $R_v^2 = \mathcal{T}'$, $R_v^1 = \mathcal{T}'$, and $R_v^0 = \mathcal{T}'$. Let $\overline{Q} = \max\{Q_v : v \in \mathcal{V}\}$. The special cases can be derived as follows:

$$\sum_{v\in\mathcal{V}} \left(\sum_{j\in\mathcal{J}\cup\{o(v)\}} \sum_{t\in\mathrm{T}} x_{jiv,t-T_{jiv}} + w_{iv,l-1} + o^{\mathrm{B}}_{iv,l-1} \right) \geq \left\lceil \frac{\sum_{t\in\mathrm{T}} D_{it} - S_i^{\max} + S_i^{\min} - \overline{\alpha}_i^{\max}}{\overline{Q}} \right\rceil \tag{3.62}$$

$$\sum_{v\in\mathcal{V}} \left(\sum_{t\in\mathrm{T}} o^{\mathrm{A}}_{ivt} + o^{\mathrm{B}}_{iv,l-1} \right) \geq \left\lceil \frac{\sum_{t\in\mathrm{T}} D_{it} - S_i^{\max} + S_i^{\min} - \overline{\alpha}_i^{\max}}{\overline{Q}} \right\rceil \tag{3.63}$$

$$\sum_{v\in\mathcal{V}} \sum_{t\in\mathrm{T}} o_{ivt} \geq \left\lceil \frac{\sum_{t\in\mathrm{T}} D_{it} - S_i^{\max} + S_i^{\min} - \overline{\alpha}_i^{\max}}{F_i^{\max}} \right\rceil \tag{3.64}$$

Constraints (3.62) impose a minimum number of arrivals at discharging port i, while Constraints (3.63) impose a minimum number of discharging operations. Finally, Equation (3.64) impose a minimum number of operations at port i in the interval $[l, k]$.

For considering the valid inequalities for the FCNF model, we replace variable o^{B}_{ivk} by variable w^{B}_{ivk} in constraints (3.53), variable $o^{\mathrm{B}}_{iv,l-1}$ by variable $w^{\mathrm{B}}_{iv,l-1}$ in constraints (3.62). The valid inequalities (3.54) and (3.64) remain the same, while constraint (3.63) is not considered as it is specific for the FCNF+ model and is redundant for the FCNF.

For using the valid inequalities for the TS model, we replace variable o^{B}_{ivk} by variable w_{ivk} in constraints (3.53), and remove variable $o^{\mathrm{B}}_{iv,l-1}$ in constraints (3.62). The valid inequalities (3.54) and (3.64) remain the same, while constraint (3.63) is not considered.

3.5 Additional Constraints

Two constraint sets for reducing the search space can be derived based on realistic assumptions. They can provide a tightening of the lower bounds depending on the instance characteristic, also making it possible to obtain better integer solutions. The first set is named as *operate-and-depart constraints*. It is based on the following assumption: Let us consider a small vessel for which capacity $Q_v \leq F_i^{\max}$ for some $i \in \mathcal{J}$. Therefore, this vessel can perform a full load or discharging operation at port i in just one time period, and after can depart to another port (of a different type) or end its route. This assumption can be incorporated into the models by introducing the following constraints:

$$\sum_{j \in \mathcal{J} \cup d(v): \Delta_i \neq \Delta_j} x_{ijvt} = o_{ivt}, \quad \forall i \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} : Q_v \leq F_i^{\max}. \qquad (3.65)$$

In TS, constraints (3.65) prevent a vessel from waiting at a port after finishing its operation (already granted in FCNF) and ensure that the vessel operates in only one time period for both formulations. These constraints may cut a possible optimal solution where a small vessel may operate more than once in a port visit or split its load, operating consecutively in two ports of the same region. However, it is not desirable to operate small values as each operation has an additional value incurred in the objective function. Also, fractioning a vessel inventory between two or more ports in a region implies that more operations will be necessary at such ports by other vessels, which should be avoided to reduce port congestion.

The second set of constraints named as *Two-port-with-no-revisit* constraints is based on the assumption with the same name proposed in Papageorgiou et al. (2014a). It assumes that when a vessel arrives in a port in some region, then: i) it will visit at most one more port before leaving the region; ii) once it leaves the port, this port will not be revisited by the vessel before leaving the region. Such constraints may cut an optimal solution where a vessel needs to visit three or more ports in the same visit to a determined geographical region. However, in the experiments reported by Papageorgiou et al. (2014a), the two-port-with-no-revisit constraints are not violated in the solutions even they are not considered in the mathematical model. To build the constraints, Papageorgiou et al. (2014a) developed an augmented time-space network that implements this assumption. However, implementing the assumption directly on the time-space and FCNF models requires additional sets of binary variables and side constraints that substantially increase the model's size, making it more challenging to solve. Thus, the following simplified constraints set is used:

$$\sum_{i \in \mathcal{J}: \mathcal{R}_i = r} \sum_{j \in \mathcal{J}: \mathcal{R}_i = \mathcal{R}_j} \sum_{t \in \mathcal{T}} x_{ijvt} \leq \sum_{i \in \mathcal{J}: \mathcal{R}_i = r} \sum_{j \in \mathcal{J}: \mathcal{R}_i \neq \mathcal{R}_j} \sum_{t \in \mathcal{T}} x_{ijvt}, \quad \forall v \in \mathcal{V}, r \in \mathcal{R} \qquad (3.66)$$

In Eq. (3.66), $\mathcal{R}_i$ denotes the region that port i belongs. The constraints ensure that the number of selected intra-regional arcs (representing the traveling arcs between ports of the same region) of a vessel in a region will be less or equal to the number of entering arcs in this region. Constraints (3.66) are effective when a unique visit of vessel v to region $r \in \mathcal{R}$ is performed along the entire planning horizon. However, suppose that more than one visit to a region occurs by the same vessel. In this case, there may exist a visit that uses no intra-regional arcs (a vessel arrives at some port in the region, operates, and departs to another region), and a second visit that uses more than one intra-regional arc, violating the assumption but not constraints (3.66).

4 SOLUTION METHODS

This chapter discusses the proposed matheuristic and metaheuristic solution methods for solving the MIRP described in Chapter 3. Firstly, it presents a pre-processing and simplification procedure to reduce the model's total number of variables and constraints in sections 4.1 and 4.2. Such procedures are also considered in the metaheuristic approach indirectly.

The matheuristic framework consists of a relax-and-fix algorithm and a fix-and-optimize algorithm, descirbed in sectionn 4.3 and 4.4. The metaheuristic approach is composed by a multi-start algorithm, a large neighborhood search, and a reduced MIP approach, described in sections 4.5, 4.6 and 4.7.

4.1 Preprocessing

Let us consider ports i and j of the same type, i.e. $\Delta_i = \Delta_j$. According to the assumption incorporated in the FCNF and FCNF+ models, a vessel always operates at a port before departing to another one. Since a minimum amount of product must be loaded or discharged when an operation takes place on the port (constraints (3.15)), then a direct voyage of vessel $v \in \mathcal{V}$ between ports i and j is impossible if $F_i^{\min} + F_j^{\min} > Q_v$ as it implies in violating the minimum operation capacity constraints. Thus, we can remove from the model the variables representing traveling arcs from i to j and vice-versa, as they cannot be used in a feasible solution.

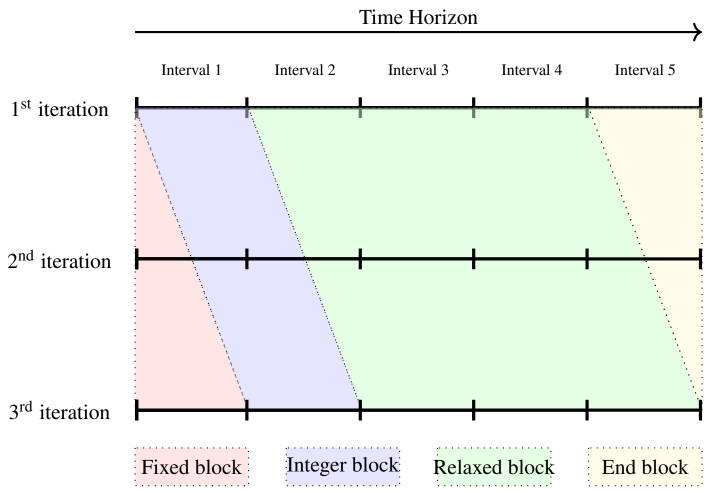
4.2 Instance Restrictions

The restriction procedure removes traveling arcs with a low chance of being used in high-quality solutions. Consider ports i and $j \in \mathcal{J}$ with $\Delta_i = \Delta_j$ belonging to different geographical regions. We know that the vessel capacity will be divided between ports i and j if a vessel travels directly from i to j and vice-versa. However, the traveling costs between regions are high due to the problem characteristic (deep-sea). If just a fraction of the vessel capacity will be used by i and j, the voyage has a higher cost per product unit, and thus it is not desirable. Therefore, we can remove all variables for each vessel representing traveling arcs between ports of the same type but belonging to different geographical regions. Although the simplification procedure may cut-off a possible integer optimal solution for the problem, computational tests have shown that no arc with the characteristics above are included in the solutions returned by our algorithms when simplification is not considered.

4.3 The Relax-and-Fix Algorithm

The proposed relax-and-fix algorithm (R&F) is based on the solution method proposed by Uggen, Fodstad and Nørstebø (2013). It consists of dividing the problem into p smaller subproblems representing an interval of the time horizon with size $\frac{|\mathcal{T}|}{p}$. Figure 4.1 illustrates the R&F along the first three iterations.

Figure 4.1: Illustration of the relax and fix algorithm along the three first iterations.



Source: from the author (2020).

According to Figure 4.1, at the first iteration the integrality constraints on all binary variables are relaxed (*Relaxed block*), except for the variables belonging to the "Interval 1", which covers the planning horizon $T = \{1, \ldots, \frac{|\mathcal{T}|}{p}\}$, and belongs to the *Integer block*. When the original problem has a time horizon substantially large, a certain number of intervals can be omitted from the model at the first iterations. These intervals belong to the called *End block*. A mathematical solver then solves the problem. At the second iteration, the blocks "move forward": binary variables of the integer interval are fixed to their obtained values and belong to the *fixed block* (original problem continuous variables are kept unfixed). A part of the relaxed block belongs now to the integer block, and a part of the model that was omitted is now considered in the model as relaxed block. The problem is solved again by the MILP solver. The algorithm continues to iterate until all intervals have been removed from the end block, and integrality constraints are reintroduced to all intervals' variables. At this point, a solution to the original problem is obtained.

We also evaluated a R&F version in which the subproblems are associated with a set composed of one or more vessels. Thus, the R&F iteratively re-insert the integrality constraint on the variables corresponding to a vessel set. The end block can be composed of a set of vessels that are not available at the beginning of the algorithm. However, with such an approach, the infeasibility degree in the solutions was higher, demonstrating that the time decomposition is most suitable for this MIRP variant.

4.3.1 Stop Criteria

In the R&F, solving each interval up to optimality does not necessarily lead to a good solution for the original problem. Thus, we use MIP relative gap (between the current integer solution and the linear relaxation at the current branch-and-bound tree node) and the time limit as stopping criteria in each iteration. As proposed by Uggen, Fodstad and Nørstebø (2013), the MIP relative gap is set initially to a positive value, which decreases linearly along with the iterations such that its value is set in the last one to 0.01%. With this strategy, the algorithm may obtain a good solution but with less processing time. It can avoid exhaustively searching for high-quality solutions at the first iteration while avoiding accepting the first feasible solution at the last iterations. The MIP relative gap of the integer solutions tends to decrease along with the iterations due to the variables fixing.

4.3.2 Avoiding Infeasibilities

During the R&F iterations, an infeasible solution can be produced due to the previously fixed decisions and the integrality constraints' re-insertion. One strategy to avoid such an infeasibility is to consider an overlap between the intervals, fixing only part of the solved integer variables (POCHET; WOLSEY, 2006). The size of the overlap is defined as a percentage value of the size of a time interval. Unlike what was done by Uggen, Fodstad and Nørstebø (2013), which applies overlap only between the fixed and the integer block, we consider it between all blocks types to keep the size of the subproblems relatively equal. As the overlap percentage increases, the tendency to produce infeasible solutions decreases. However, high overlap levels increase the processing time of the algorithm.

Even using overlap, port-time inventory bounds (3.16) can be violated if no vessel can operate at a port in specific times, and variables α_{it} are insufficient to avoid stockout or inventory overflow. To handle this issue, nonnegative auxiliary variables $\beta_{it}, \theta_{it}, i \in \mathcal{J}, t \in \mathcal{T}$ are introduced to the model, as proposed by Friske and Buriol (2017), Friske and Buriol (2018). They turn the port-time inventory constraints into soft constraints, being their use penalized in the objective function. More precisely, these variables' coefficient is set to 1000, which is sufficiently larger for the tested instances. The coefficient value can also be defined as $2 \times \max_{i \in \mathcal{J}, j \in \mathcal{J}, v \in \mathcal{V}} \{C_{ijv}\}$. Constraints (3.5) are reformulated as follows:

$$s_{it} = s_{i,t-1} + \Delta_i (D_{it} - \sum_{v \in \mathcal{V}} f_{it}^v - \alpha_{it} - \beta_{it} + \theta_{it}), \ \ \forall i \in \mathcal{J}, t \in \mathcal{T} \tag{4.1}$$

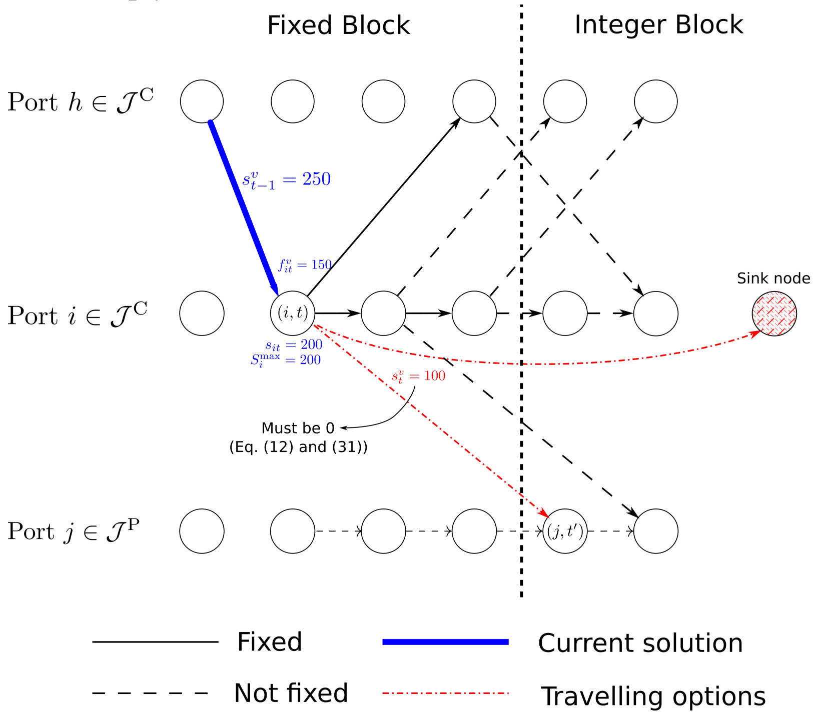
The auxiliary variables on constraints (4.1) prevent the solver from stopping prematurely during the iterations. However, if some auxiliary variable is positive at the end of the R&F, the original problem's solution remains infeasible.

Another kind of infeasibility can occur when flow balance constraints are inconsistent with traveling at capacity and traveling empty constraints. An example is illustrated in Fig. 4.2.

In Figure 4.2, it is fixed the decision of vessel v to travel from port h to port i, as well as the decision to operate at node (i, t). The amount f_{it}^v discharged at node (i, t) is sufficient to bring the inventory to its maximum capacity. Thus, there is still load onboard at vessel v, and its unique option is to travel to the sink node or the loading port j, as the decision variables associated with the waiting arcs and traveling arcs to ports of the same type are fixed to zero. This is infeasible, as a vessel cannot travel from a discharging port to a loading port (or end its route) with some load on board (equations (3.12) and (3.40)). This infeasibility can be handled by θ_{it} variables, allowing vessel v to discharge all its load without violating the port's inventory capacity. However, if the amount to be discharged by vessel v in one time period is larger than the operation capacity $F_i^{\max}$, the solution will remain infeasible. Such infeasibility can also occur in cases where port i is a loading port, violating the traveling at full capacity constraints (equations (3.11) and (3.39)).

The higher probability of this infeasibility occurs when considering small-sized

Figure 4.2: Example of infeasibility that may occur using the R&F by violating travel at capacity and travel empty constraints.



Source: From the author (2020).

intervals combined with higher levels of overlap in the R&F algorithm and considering the preprocessing strategy (Section 4.1).

4.3.3 Speeding up Computation

To avoid repeatedly solving the linear relaxation of the variables in the relaxed block, saving up processing time, Uggen, Fodstad and Nørstebø (2013) introduced the **end block**, which is derived from the last part of the relaxed block. The end block can be a horizon cut or a so-called light model. The light model simplifies the subproblem based on specific problem knowledge, while the horizon cut ignores part of the planning horizon. Uggen, Fodstad and Nørstebø (2013) observed that there is no significant difference between the horizon cut and light model concerning objective value. However, the horizon cut provides better processing times. On the other hand, the solution infeasibility increases as the size of the end block increases.

We consider in our work to use the horizon cut strategy in the end block. We propose two modifications to the subproblems' parameters to reduce the drawback of the horizon cut. The first one establishes a proportional value of $\alpha_i^{\max}$ according to the number of time periods in the current subproblem. Let e be the number of time periods in the first iteration of R&F, i.e., not considering the time periods in the end block. We then define $\alpha_i^{\max} = \frac{\alpha_i^{\max}}{|\mathcal{T}|} \cdot e$. At each iteration, the value of $\alpha_i^{\max}$ is updated proportionally to the number of time periods considered in the subproblem until there is no interval in the end block and the parameter is set to its original value. This modification tries to avoid scenarios where variables α_{ij} reach their maximum cumulative limit before the whole planning horizon is revealed. Therefore if new α_{it} variables need to take positive values, no feasible solution will be possible.

The second modification tries to prevent possible inventory violations in the ports when part of the planning horizon is revealed to the subproblem. This behavior is similar to the end of horizon effects described by Agra et al. (2017), where the inventories can get close to their limits at the end of the planning horizon, resulting in a more difficult problem in the next planning operation. For avoiding such situations, in the last time period of the current subproblem, we reduce the total capacity of the loading ports by 10% and impose that the minimum inventory at the discharging ports is 10% of its total capacity.

4.3.4 Updating Valid Inequalities

Valid inequalities (3.53)-(3.54),(3.62)-(3.62) use the initial inventory S_i^0 when $l = 0$, which provides strongest knapsack cuts. We can extend such cuts in the R&F algorithm for other time periods when index l belongs to the fixed block. At each R&F iteration, the inequalities in which l belongs to the fixed interval have their right-hand side parameter $S_{il}^{\min}$ (loading ports) or $S_{il}^{\max}$ (discharging ports) updated to the current inventory s_{il}.

4.4 Fix-and-optimize Algorithm

The fix-and-optimize algorithm (F&O) starts after a solution is obtained by the R&F. It tries to improve parts of the current solution iteratively. The method consists of fixing all integer variables from the solution obtained by the R&F, and iteratively a set of these variables is unfixed to be optimized by the solver, generating different subproblems. Continuous variables are always free to be optimized. Four rules or strategies for selecting the set of variables to be optimized are defined:

1. **Time Intervals (TI):** Consists of dividing the time horizon into m intervals, as in the R&F. At each iteration, the integer variables of one interval are unfixed. After optimized, these variables are fixed to the newly obtained values. This procedure is repeated iteratively until no improvement is achieved by optimizing at least one interval in m iterations. The strategy is used in the works of Uggen, Fodstad and Nørstebø (2013), Lang and Shen (2011). We also consider an overlap value u^{TI} between the intervals defined as a parameter, based on the work of Uggen, Fodstad and Nørstebø (2013).

2. **Vessels Pairs (VP):** This is strategy is the same proposed by Goel et al. (2012) which explores the neighborhood between two vessels, this procedure consists of iteratively selecting a pair of vessels to be optimized. Let v_1 and v_2 be the vessels selected to be optimized at some iteration. Then, all binary variables indexed by $v \in \mathcal{V} : v = \{v_1, v_2\}$ are unfixed, and the solver is started. The vessel pairs can be selected at random with no repetitions, or in a lexicographic order, i.e. $(v_1, v_2), (v_1, v_3), \ldots, (v_{|\mathcal{V}|-1}, v_{|\mathcal{V}|})$. The search stops when no improvement on the solution is obtained in the last $\binom{|\mathcal{V}|}{2}$ iterations.

3. **Interval Vessel (IV)** This strategy is proposed in this work and can be considered a combination of the two previous methods, iterating between time intervals and vessels. The time horizon is divided into m intervals, allowing one interval to be optimized at a time. Also, all integer variables associated with one vessel are allowed to be optimized. After solving the subproblem, the optimized vessel's integer variables are fixed to the newly obtained values, except those belonging to the interval which is being optimized, and a next vessel is selected to be optimized. When all vessels were optimized, the current interval is fixed to their current solution, and the next interval is selected. The algorithm iterates between all time intervals and all vessels, corresponding to $m|\mathcal{V}|$ steps in a complete iteration. The search stops when no improvement is achieved in one complete iteration. As in the time intervals strategy, we can define a percentage overlap u^{VTI} between the intervals, giving a total of $\lceil m \cdot (1 + \frac{u^{\mathrm{VTI}}}{100}) \rceil \cdot |\mathcal{V}|$ subproblems, assuming that an overlap of 100% is not allowed.

4. **Port Types (PT)** This procedure is based on the work of Papageorgiou et al. (2014a). First, all integer variables associated with the loading ports are fixed, while integer variables associated with the discharging ports are optimized. Then, variables of

the discharging ports are fixed to the new values, and the variables of the loading ports are optimized. Variables that correspond to arcs that connect ports of different types are kept unfixed in the whole procedure to allow a vessel to depart from a region earlier if possible. According to Papageorgiou et al. (2014a), optimizing first discharging ports is justified since the instances usually have more discharging ports than loading ports. This procedure repeats until no improvement is achieved in solving the two ports type consecutively.

Besides the stopping criteria of each improvement approach, each iteration/step has a time limit and a MIP relative gap as stopping criteria.

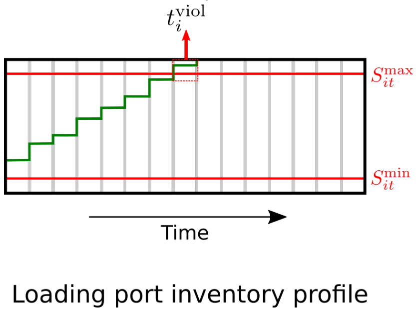
In more recent experiments (described in Section 5.3.4-5.3.7)), before starting the F&O, the operate-and-depart constraints, the valid inequalities, and the slack variables (if not used) are removed from the model as they increase its size and are relevant to obtain feasible integer solutions only. Also, since the operate-and-depart constraints (Section 3.5) can cut an optimal solution, it may be relevant to remove them to allow a better exploration of the solution space by the F&O. The same applies to the modified valid inequalities (Section 4.3.4) as they generate local cuts along with the R&F iterations.

4.5 The Multi-start Algorithm

The multi-start algorithm builds a set S of n solutions s using a constructive greedy heuristic with randomness. For explaining the heuristic, the following notation is adopted:

- $\mathcal{R}_i$ - Set of ports corresponding to the geographical region of port i.
- t_i^{viol} - is the earlier time period t in which an inventory violation occurs at port i, i.e. $t_i^{\text{viol}} = \min\{t \in \mathcal{T} : s_{it} < S_i^{\min}, i \in \mathcal{J}^{\text{C}}\}$, and $t_i^{\text{viol}} = \min\{t \in \mathcal{T} : s_{it} > S_i^{\max}, i \in \mathcal{J}^{\text{P}}\}$. It is illustrated in Figure 4.3;
- *Vessel voyage* - Sequence of vessel actions starting by departing from port $i \in \mathcal{J}$, and ending at port $j \in \mathcal{J}$, such that i and j are different type ports. Optionally, a port $i' \in \mathcal{R}_i$ can be visited after port i, and a port $j' \in \mathcal{R}_j$ can be visited after port j. A vessel voyage ends after finishing its operation at port j (or j'), such that no more operations can be performed, i.e., if $j \in \mathcal{J}^{\text{P}}$ the vessel is at full capacity, and if $j \in \mathcal{J}^{\text{C}}$ the vessel is empty.

Figure 4.3: Inventory violation time t_i^{viol}.



Source: From the author (2020).

The constructive algorithm iteratively builds a concatenation of voyages for each vessel until there is no inventory violation at the ports, i.e. $t_i^{\text{viol}} > |\mathcal{T}|, i \in \mathcal{J}$.

The multi-start algorithm and its main components are presented in Algorithm 3 and detailed in the next sections. Parameters p^{CP} and p^{2nd} represent probabilities for selecting ports at random along with the algorithm iterations.

Algorithm 3: Multi-Start Algorithm

```
1:  Input: n, p^CP, p^2nd
2:  Output: a set of solutions S
3:  while |S| < n do
4:      Init(s, p^2nd)
5:      while t_j^viol ≤ |T|, i ∈ J do
6:          j = urgentPort()
7:          i = counterpartPort(j, p^CP)
8:          v = selectBestVessel(i, j, p^2nd)
9:          UpdadeSolution(s, v, i, j)
10:     end while
11:     CompleteSolution(s)
12:     S = S ∪ s
13: end while
```

4.5.1 Initializing Solution s

This function presented in line 4 puts each vessel $v \in \mathcal{V}$ in its initial port j_v^0 and its first time available t_v^0 given as parameter input. We assume that vessels always operate at the initial port, as they start at full capacity ($s_0^v = Q_v$) if $j_v^0 \in \mathcal{J}^{\text{C}}$ and start empty ($s_0^v = 0$) if $j_v^0 \in \mathcal{J}^{\text{P}}$. The algorithm decides if vessel v will also operate at a port $j' \in \mathcal{R}_{j_v^0}$ after operated at j_v^0. In this case, it is necessary to split the vessel capacity between ports j_v^0 and j'. This decision procedure is detailed in Section 4.5.6. For now, assume that after the end of the initialization function, all vessels $v \in \mathcal{V}$ operated at its initial port j_v^0 (and possibly at a port $j' \in \mathcal{R}_{j_v^0}$), completing its first *vessel voyage*.

Different vessels may start the route at the same port but in different time periods. Thus, the order in which each vessel is initialized in the solution may affect either the port inventory levels and the other vessels' operations at the same port. For obtaining more variability in the solutions, this order is randomly defined by the algorithm for each solution $s \in S$.

After a vessel operation is implemented at port $j \in \mathcal{J}$, the inventory violation time t_j^{viol} is updated, incresing its value.

4.5.2 Selecting the Urgent Port

After the solution was initialized, the urgent port j is selected (line 6) as the port with the earliest inventory violation time, i.e., $j = \arg\min_{i \in \mathcal{J}}\{t_i^{\text{viol}}\}$. It can be a loading or a discharging port, and in case of a tie, the discharging port is selected. If the tie occurs between ports of the same type, the one with the highest inventory capacity is selected.

4.5.3 Selecting the Counterpart Port

The counterpart port i is selected for supplying the demand of urgent port j (line 7). The candidate ports are always of a different type of port j, i.e., $\Delta_i \neq \Delta_j$. They are ranked in increasing order considering t_i^{viol} and the distance to the urgent port j. The port with the minor sum of its position in the two ranks is selected as the counterpart port, and in case of a tie, the port with the earliest inventory violation time is selected. This strategy is the same as proposed by (CHRISTIANSEN et al., 2011). Additionally, we defined a probability p^{CP} of selecting a counterpart port at random between all candidate ports to introduce more variability on the solutions generated.

4.5.4 Selecting the Best Vessel

The function in line 8 evaluates each vessel $v \in \mathcal{V}$ to perform the voyage between counterpart port i and urgent port j. For the explanation, we assume that i is a loading port, and j is a discharging port. If v is currently at a loading port $l \neq i$ in the partial solution, it does not need to visit port i, as v will be already loaded and can travel directly to port j. Otherwise, v needs to travel from its current position to the counterpart port i. In such case, it is also evaluated if a port $i' \in \mathcal{R}_i$ should be visited after port i and before departing to urgent port j (see section 4.5.6). After visiting and operating at port i (and possibly at port i'), vessel v travels at full capacity to urgent port j. It is also evaluated if a port j', such that $\mathcal{R}_{j'} = \mathcal{R}_j$ can be visited after j. Note that when evaluating each vessel, the optional ports i' and j' can be different.

Sometimes a vessel can arrive at a port only after t_i^{viol}. In such cases, spot market variables α_{it} are used to keep the inventory at the bound limits ($s_{it} \leq S_i^{\max}$ for loading ports, $s_{it} \geq S_i^{\min}$ for discharging ports) until one time period before the vessel can arrive and operate at the port. As the use of spot market variables is limited, if they are not sufficient to keep the inventory of ports at the bound limits until the vessel arrives, auxiliary variables $\beta_{it}, i \in \mathcal{J}, t \in \mathcal{T}$ are used. They are equivalent to α_{it}, but not bounded and highly penalized in the objective function as in the R&F (see section 4.3.2). The costs of using α_{it} and β_{it} variables are considered penalization costs of the voyage.

After vessels were evaluated, the best one is selected according to one of the following criteria, chose at random:

1. *Lowest cost*: Considers the traveling, operations, and possible penalization costs minus the profit obtained by the vessel in the voyage;

2. *Highest cost*: The opposite of the previous criteria;

3. *Smallest number of waiting times*: Considers the total number of time periods in which a vessel needs to wait in the ports along the voyage;

4. *Closest arrival time*: Calculates the difference between the arrival time and the inventory violation time at the urgent port. The lower the value, the better;

5. *Earliest arrival time*: Selects the vessel which can first arrive at urgent port j to operate;

6. *Smallest number of operations*: Selects the vessel which needs the lowest number of operations at the ports to finish the voyage;

7. *Lowest cost/capacity ratio*: considers the vessel voyage cost divided by its capacity Q_v;

8. *Lowest penalization*: in cases when no vessel can arrive at urgent port j before inventory violation time, selects the vessel with the lowest penalization cost associated.

Criteria 1, 3, 4, 6–8 aims to improve solutions quality, while criteria 2 and 5 aims to provide variability on the generated solutions.

While the constructive algorithm iterates, it is possible that a vessel is able to operate in a port at a time period earlier than an already implemented operation. In such cases, for defining a feasible operation, it is necessary to check the inventory of the port at the last time period in which the operation took place. For example, consider that an operation was implemented in port $j \in \mathcal{J}^{\mathrm{C}}$ at time $t' = 9$, such that $s_{jt'} = 250$. The port inventory capacity is $S_j^{\max} = 300$. Now suppose that vessel v arrives in port j at time $t = 4$ and $s_{jt} = 100$. Considering only the inventory at time t, vessel v can discharge 200 units without violating the inventory capacity of port j. However, at time t', the maximum amount that can be discharged is only 50 units. Therefore, if some discharging operation occurs before time t', the value to be discharged must be at most the inventory space available at time t'. The following variable parameter defines the maximum value that can be operated at port i in time t without causing an inventory violation in future time periods:

$$f_{it}^{\max} = \begin{cases} \min_{t' \in \mathcal{T}: t' \geq t}\{s_{it'}\} & \text{if } i \in \mathcal{J}^{\mathrm{P}} \\ \min_{t' \in \mathcal{T}: t' \geq t}\{S_i^{\max} - s_{it'}\} & \text{if } i \in \mathcal{J}^{\mathrm{C}} \end{cases} \tag{4.2}$$

4.5.5 Specific procedures

Two problem specific functions are used internally in $Init$ and $selectBestVessel$ functions. For explaining them, we consider a discharging port $i \in \mathcal{J}^{\mathrm{C}}$. The procedure for a loading port is analogous.

4.5.6 Deciding between operating at one or two ports

When evaluating a vessel v to discharge at port i, the algorithm must decide between discharging all its cargo in port i or splitting the vessel load with a second port $i' \in \mathcal{R}_i$.

Such a decision is used to handle two issues: i) no other vessel except v can arrive at port i' before $t_{i'}^{\text{viol}}$. Therefore, it is preferable that v visits port i' to avoid a major probability of building an infeasible solution; ii) vessel v needs to wait too many time periods to discharge in port i due to inventory constraints. Although there are no costs for a vessel to wait at a port, such a situation should be avoided as it can compromise the next voyages. Thus, it is possible that dividing a vessel load between two ports may result in lesser waiting time periods, and the voyage can finish earlier.

The algorithm ranks all candidate ports $i' \in \mathcal{R}_i$ (including port i) considering three criteria: inventory violation, end time of the operation, and voyage cost. Firstly it is evaluated if only vessel v can arrive at a candidate port i' before $t_{i'}^{\text{viol}}$. If this is the case, vessel v must visit port i'. If there is no candidate port in such a situation (or more than one), the algorithm selects the port i' in which vessel v ends its voyage earlier in the planning horizon. If there is a tie between two or more candidate ports considering this criterion, the port i' that implies in the lowest voyage cost is selected.

If the best-ranked port is different from i, the vessel capacity is divided between the ports such that the maximum amount is operated in port i at time t, and the remaining is operated at port i', respecting the minimum amount $F_{i'}^{\min}$. Otherwise, if $i' = i$, the total vessel capacity is operated at port i. Besides the ranking, we defined a probability p^{2nd} of randomly selecting a port $i' \in \mathcal{R}_i$ to provide more variability in the solutions.

4.5.7 Defining operation values and times

Supose that vessel v arrives in port i at time t, and needs discharge the amount f. This operation is subject to several problem constraints, such as inventory, operation, and berth limits. Firstly, if berth occupation in i at time t is equal to berth limit B_i, then vessel v must wait at the port until a time $t' \geq t$ in which a berth is available. There are three options for a vessel to discharge in a port: the first and most preferable is to discharge f in port i at time t without waiting at the port, i.e., $f_{it}^v = f$. This is possible only if i has capacity to recieve the amount f, and there is sufficient inventory space at the time t, i.e., $f \leq F_i^{\max}$ and $f \leq S_i^{\max} - s_{it}$. In the second option, the vessel needs to wait one or more times while $f > S_i^{\max} - s_{it}$ before discharging f. The third option occurs when $f > F_i^{\max}$. In such a case, a vessel must operate multiple time periods, splitting the amount f.

4.5.8 Updating Solution

This procedure (line 9 of Algorithm 3) implements in solution s the voyage of the selected vessel v between ports i and j, updating their inventory according to the amount operated and also the inventory violation time t_i^{viol} and t_j^{viol}. The port inventory is updated for each time period, starting from the operation time until the end of the planning horizon. Thus, the operation's complexity time is $O(|\mathcal{T}|)$ for each port visited in the vessel voyage.

4.5.9 Complete the Solution

After the end of the while loop, there is no port with an inventory violation, and a feasible solution is obtained (unless some auxiliary variable β_{it} is positive). The *CompleteSolution* function in the line 11 aims to obtain a better solution quality, verifying for each discharging port if there is a vessel that can discharge in the port before the end of the planning horizon. If such a case exists and the operation provides a revenue greater than the voyage and operation costs, the voyage is performed.

4.6 The Large Neighborhood Search

The large neighborhood search is responsible for improving solutions given by the multi-start algorithm and possibly remove the infeasibility when some variable β_{it} is positive. The LNS partially destroys a solution s by removing $u\%$ of the vessels at random. Each removed vessel's route is then rebuilt in a different way, generating a candidate solution s'. If solution s' is accepted, it becomes the current solution s. The procedure is repeated until it reaches a maximum number of iterations $maxIt^{\text{LNS}}$, or the candidate solution is not accepted for $\max^{\text{NA}}$ consecutive iterations. The best solution found is returned at the end of the LNS.

The route of vessels is rebuilt with the same constructive heuristic of the multi-start algorithm, given in lines 5-11 of Algorithm 3. The unique difference is that the constructive heuristic in the LNS considers only the vessels removed from the solution. The LNS is performed on a set $S^{\text{LNS}} \subseteq S$ which is composed of m solutions, such that $\lceil \frac{m}{2} \rceil$ are the best solutions obtained by the multi-start algorithm (considering different objective function values) and $\lfloor \frac{m}{2} \rfloor$ are solutions chosen at random from set S. This configuration is chosen as we observed that a significant part of the multi-start algorithm's infeasible solutions could be turned feasible, besides obtaining a lower objective function value than the best multi-start solutions improved by the LNS.

4.6.1 Acceptance Criteria

A candidate solution s' is accepted if its objective value is better than the objective value of the current solution s. The algorithm also accepts worse solutions than the current one under some conditions. To do this, we use the temperature parameter used in simulated annealing in a similar way as (ROPKE; PISINGER, 2006), in which the probability of a worse solution being selected decreases as the temperature downs. More precisely, the probability is given by $p(S) = e^{-\frac{f(S')-f(S)}{T}}$ where T is the temperature. The initial temperature is defined by the objective value of the initial solution (provided by the multi-start algorithm) multiplied by a factor $0 < f^{\text{start}} < 1$, and decreased at each iteration by a cool rate $0 < c < 1$, until reaching the final temperature defined based on the initial solution objective and a constant $0 < f^{\text{end}} < 1$, such that $f^{\text{end}} < f^{\text{start}}$.

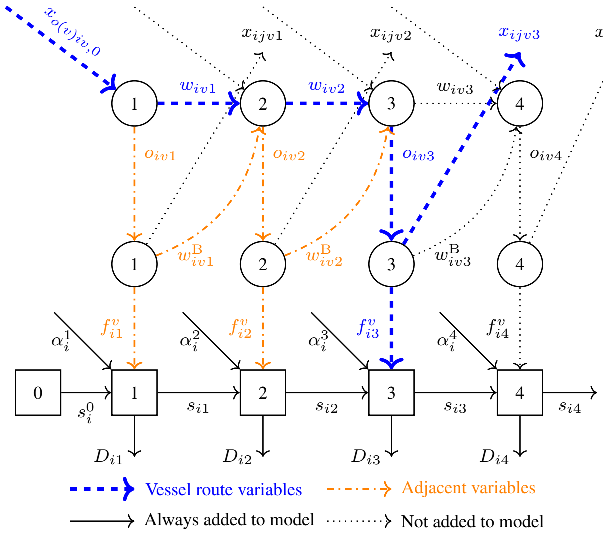
4.6.2 Backtracking

To avoid the LNS getting stuck in local minima and intensify the search in promising solution spaces, if a new best-solution is not found after max^{reset} iterations, the current solution is set to the best solution found so far. The current temperature is set to half of the starting temperature to provide more exploration of the search space from the best solution.

4.7 The reduced MIP

After LNS was performed on the m solutions of set S^{LNS}, such solutions are converted to a reduced mixed-integer program, based on the FCNF model. In the reduced MIP only the variables (and respective constraints) that assume a positive value in the corresponding metaheuristic solution are inserted in the model. The exceptions are port inventory variables s_{it}, spot market variables α_{it}, and auxiliary variables β_{it}, which are added for all $i \in \mathcal{J}, t \in \mathcal{T}$. A mathematical solver then solves the reduced model for

Figure 4.4: Example of a solution in the FCFN model and the corresponding adjacent variables.



Source: From the author (2020).

t^{RMIP} seconds for trying to obtain a better solution than the best one provided by the LNS. As the size of the reduced model (number of columns and rows) is smaller than the original model, it can be solved to optimality in a short processing time depending on the value of parameter m.

Although solving RMIP is usually quick, adding only the variables and constraints corresponding to the LNS solutions can be restrictive so that the solution found by the solver is the same found by the LNS. Thus, we defined “adjacent" variables of the solution that are also included in the model. They cover the binary variables o_{ivt}, w_{ivt}, and w^{B}_{ivt}, besides their corresponding continuous flow variables f^{v}_{it}, f^{O}_{ivt}, and f^{WB}_{ivt}, respectively. Figure 4.4 illustrates the adjacent variables according to a vessel v route solution visiting a port i.

In the example of Fig. 4.4, vessel v arrives in port i at time $t = 1$, wait for three time periods, and then operates at time $t = 3$, finally departing to another port. The adjacent variables correspond to the waiting (w) and operation (o) variables between the arrival time $t = 1$ of vessel v in port i until the departing time $t = 3$. They enable the solver to decide if a vessel can improve its route by changing possible time periods in which the operation(s) take(s) place and the value which is operated. Note that in the example of Fig. 4.4, vessel v still must operate at the time period $t = 3$, as the traveling arcs variables x_{ijvt} representing the depart from port i in time periods with $t < 3$ are not included in the model.

5 COMPUTATIONAL RESULTS

This chapter presents the computational results obtained by solving the MIRP using the matheruistic framework and the metaheuristic approach. Firstly, the benchmark instances used in the experiments are presented in Section 5.1. After, a lower bound analysis to compare the TS and FCNF models is presented in Section 5.2. Finally, Section 5.3 presents the results and analysis of the matheuristic framework, followed by the metaheuristic results presented in Section 5.4.

The integer solutions were compared with the state-of-the-art algorithm for the presented MIRP, proposed by Papageorgiou et al. (2014a). For comparison purposes, the models were solved as minimization problems, inverting the objective function coefficients' signal.

The algorithms were implemented in C++ and compiled with the optimization parameter $-\mathrm{O3}$. The experiments were carried out in different computer configurations, and different versions of the CPLEX solver were used. We detail them in each section.

The valid inequalities presented in section 3.4 were added to the models a priori for all combinations of $\mathcal{T}$, where $\mathcal{T}'$ includes either the first period or the last period, i.e., $\mathcal{T}' = \{1, \ldots, t\}$, or $\mathcal{T}' = \{t, \ldots, T\}, t \in \mathcal{T}$. Thus, each set of valid inequalities adds to the model $\Theta(2\mathcal{J}(\mathrm{T}{-}2))$ constraints. Depending on the problem instance, some cuts may be redundant. We let the solver handle this issue.

5.1 Benchmark Instances

The computational experiments used the "Group 1" instances available in the MIRPLIB (PAPAGEORGIOU, 2013). The instances name present their characteristics. For example, in instance "LR2_11_DR2_33_VC5_V12a" there are 2 loading regions (LR), and in each region, there is one loading port, two discharging regions (DR), each of them with three ports. There are five classes (VC), with a total of twelve available vessels (V), at least one for each vessel class. The letter at the end of the name is used for differentiating instances of the same size. The benchmark set consists of 14 instances, such that the planning horizon of each instance can be changed and set up to $|\mathcal{T}| = 360$ days. The tests were performed with planning horizons of 45 and 60 time-periods for the matheuristic framework (Section 5.3). In the experiments with the proposed metaheuristic (Section 5.4), instances with 120, 180, and 360 time periods were also considered.

5.1.1 Modified Instances

A simple modification on the MIRPLIB instances is proposed for evaluating the quality of the matheuristic framework to solve more general instance cases where ports are not grouped in regions. Let x_i and y_i be the coordinates (in a Euclidean plane) of each port $i \in \mathcal{J}$ in the original instance. Also, let $\overline{x} = \max_{i \in \mathcal{J}}\{x_i\}$, $\overline{y} = \max_{i \in \mathcal{J}}\{y_i\}$, $\underline{x} = \min_{i \in \mathcal{J}}\{x_i\}$, and $\underline{y} = \min_{i \in \mathcal{J}}\{y_i\}$ be the extreme coordinates of the instance. Then, new coordinates for each port i are defined at random as follows: $x_i = rand(\underline{x}, \overline{x})$ and $y_i = rand(\underline{y}, \overline{y})$, where $rand(a, b)$ is a function that generates a random value between a and b. The seed value used for each instance was $\overline{x}$. The distances between ports and cost of arcs are recalculated according to (PAPAGEORGIOU et al., 2014b). Instance "LR2_22_DR2_22_VC3_V10a" becomes infeasible due to the new ports' positions. For this case, the values of $D_{it}, i \in \mathcal{J}, t \in \mathcal{T}$ were reduced in 10%. The modified instances are available on the author's web page[1]. The tests with the modified instances were carried out only on the TS model results of Section 5.3.1. Observe that such modification in the instances may turn the two-port-with-no-revisit constraints described in section 3.5 not valid, as they are based on scenarios that ports are grouped in geographic regions.

5.2 Lower Bounds

This section presents an evaluation of the quality of the formulations presented in Chapter 3 considering the lower bounds obtained by solving the models' linear relaxation and presenting the contribution of the valid inequalities to tight the lower bound.

The average results considering instances with $|T| = \{45, 60\}$ are presented in Table 5.1, including the lower bounds obtained by the *Optimistic SystemModel* and the *SystemModel-2Port* obtained by Papageorgiou et al. (2014a). The detailed table, including individual instance results, is available in Table A.1 in the appendix. The Optimistic SystemModel is a simplified version of the TS, which groups ports per region and vessel per class. The SystemModel-2Port is a tailor-made formulation adapted from the TS, and the results reported are obtained by solving the model for up two hours. Column *Avg. Obj.* presents the average values of the models' linear relaxation considering all instances. The lines are ordered in increasing order according to the value in column *Avg. Obj.*, i.e., from the loose to the tighter lower bound. Column *Improv.* presents the

[1] <http://inf.ufrgs.br/$\sim$mwfriske>

Table 5.1: Average lower bound values considering different formulations.

Model	Avg. LB.	Improv.
TS	-38,915.6	-
TS+VI	-30,260.4	22.2%
Optimitistic SystemModel	-16,895.6	56.6%
FCNF	-16,633.0	57.3%
FCNF + VI	-16,564.2	57.4%
SystemModel-2Port	-15,684.9	59.7%

Source: From the author (2020).

percentage improvement of the model against the base case (TS) given by $|\frac{LB^{\text{TS}}}{LB} - 1| \cdot 100$, where LB is the linear relaxation of the current model, and LB^{TS} is the lower bound of the base case. Note that we present only the lower bound of the FCNF model, as the lower bounds obtained by the FCNF+ model are the same.

According to the results shown in Table 5.1, we observed that:

- As expected, the lower bounds obtained by the FCNF model are much better than those obtained by the TS formulation;
- The use of valid inequalities contributed significantly to tight the TS model lower bounds, but they were not expressive for the FCNF model.
- The OptimisticSystem Model lower bounds are slightly worse than those produced by the FCNF model, while the lower bounds of the SystemModel-2Port improved the FCNF lower bounds by 5.6%. It is important to note that the lower bounds produced by the SystemModel-2Port were obtained from a tailor-made model considering practical problem assumptions. The FCNF model's lower bound quality is due to the formulation itself.

Although not reported in Table 5.1, the processing time for obtaining the lower bounds for the FCNF model when adding all valid inequalities a priori grown considerably. For example, for obtaining the lower bound of instance LR2_22_DR3_333_VC4_V17a with $T = 60$, the FCNF model took approximately 17 hours, and when adding the valid inequalities, it took nearly 40 hours. An alternative for reducing the computational time is adding valid inequalities to the model on-demand, using a separation problem. However, solving the linear relaxation of the pure FCNF is also time-consuming for several instances, and since the separation problem needs to solve the linear relaxation frequently, the gain in processing time may be small. Another point is that adding all valid inequalities a priori when solving the problem with the relax-and-fix algorithm does not affect the processing time. On the contrary, it reduces the total computation time, as can be observed in sections 5.3.2 and 5.3.5.

5.3 Matheuristic Results

This section presents the results obtained by the R&F and the F&O using the TS and FCNF models. It is organized as follows:

- Section 5.3.1 presents the results obtained by solving the TS model with the proposed framework and a priori parameter setting.
- Section 5.3.2 presents the results by solving the FCNF model with the proposed framework and a priori parameter setting.
- Section 5.3.4 presents the algorithms' parameter callibration using the *irace* package.
- Section 5.3.5 presents an individual analysis of the different formulation components used in the solution strategy.
- Section 5.3.6 presents the evaluation of each F&O strategy individually.
- Section 5.3.7 presents the framework results obtained with the parameter calibration for both TS and FCNF model.

Since this thesis summarizes the research along the Ph.D. period, some features of the MIRP formulations, R&F, and F&O were not considered in part of the computational experiments as they were proposed after such experiments. Table 5.2 lists the features proposed for the algorithm and formulations to summarize such differences, pointing out which of them were used in the experiments according to the section, including the publication in which the results were presented.

5.3.1 Time-space Network Model Results

The results presented in this section were obtained using the CPLEX 12.5 solver, running in an AMD-FX-8150 computer running at 3.6 GHz on a single core, with 32 GB RAM.

Table 5.2: Models, formulations features, and matheuristic features considered in the experiments.

	Model	TS	FCNF+	TS	FCNF
Section		5.3.1	5.3.2	5.3.4,5.3.5, 5.3.6, 5.3.7	
Correspondent publication		Friske and Buriol (2017)	Friske and Buriol (2018)	Friske, Buriol and Camponogara (2020)	
Formulation	Tight Flow Variables Bounds (Sec. 3.2.1)				•
	Valid Inequalities (Sec. 3.4)		•	•	•
	Operate-and-depart constraints (Eq. (3.65))	•	•	•	•
	Two-port-with-no-revisit constraints (Eq. (3.66))	•			
	Preprocessing (Sec. 4.1)			•	•
	Instance Restrictions (Sec. 4.2)			•	•
R&F	Use of auxiliary β_{it} variables	•	•	•	•
	Use of auxiliary θ_{it} variables		•	•	•
	Proportional $\alpha_i^{\max}$ (Sec. 4.3.3)			•	•
	Tight Inventory (Sec. 4.3.3)			•	•
	Updating Valid Inequalities (Sec. 4.3.4)			•	•
F&O	Overlap u^{TI} between intervals (TI)			•	•
	Overlap u^{VTI} between intervals (IV)			•	•
	Remove valid inequalities and additional constraints before start			•	•
	Remove auxiliary variables β_{it},θ_{it} before start			•	•
	Random selection of vessels in VP strategy	•	•		
	Lexicographic selection of vessels in VP strategy			•	•
Parameter setting		A priori	A priori	*irace*	*irace*

Source: From the author (2020).

For defining the parameters of the matheuristic framework using the time-space network model, the instances were divided into two sets according to the number of loading regions. Set-1 corresponds to the instances with one loading region (LR1), while set-2 corresponds to the instances with two loading regions (LR2). There is no distinction between the time horizons of the same instance, i.e., the same parameters used for $T = 45$ are also used for $T = 60$. The exception occurs with the number of intervals p that the time horizon is divided into the relax-and-fix algorithm. Parameters and possible values tested for each set are described in Table 5.3.

Table 5.3: Parameters values used in for the time-space model

	Parameter	Acronym	Value Set-1	Value Set-2	
R&F	Number of intervals	p	{5,9}		Case $T = 45$
			{6,10}		Case $T = 60$
	Overlap (%)	o	{15,30,50}		
	Time limit for solving each subproblem (s)	$t_{\text{RF}}^{\text{it}}$	{50,100,200}	{100,200,400}	
F&O	Time limit per iteration (s)	$t_{\text{FO}}^{\text{it}}$	25	50	Not using β_{it}
			35	75	Using β_{it}
			70	150	
			140	300	
	F&O time limit (s)	$t_{\text{FO}}^{\max}$	7200	10800	

Source: From the author (2020).

According to Table 5.3, each instance set can have more than one value for each parameter. The combination with the smallest value for each parameter is first tested, and when necessary, some of them are increased. For example, consider an instance from Set-1 with $T = 45$, the first test uses $p = 5$, $o = 15$, $t_{\text{RF}}^{\text{it}} = 50$. If the solution becomes infeasible during a R&F iteration, the overlap is increased from 15% to 30%, and the test is restarted. On the other hand, if R&F cannot find an integer solution in some iteration due to the time limit per iteration $t_{\text{RF}}^{\text{it}}$, it is increased from 50 to 100 seconds. Even if with the maximum values of o and $t^{\mathrm{it}}_{\mathrm{RF}}$ no integer solution was found, or solutions remained infeasible, the value of p is changed from 5 to 9, and the other parameters are reset to the minimum values, increasing them if necessary. If no solution has been found by varying the previous parameters, only the auxiliary variables β_{it} (see section 4.3) are added to the model, again resetting p, o and $t^{\mathrm{it}}_{\mathrm{RF}}$ to its minimum values. When using auxiliary variables, $t^{\mathrm{it}}_{\mathrm{FO}}$ is also increased. If a solution remains infeasible during R&F or at the end of the local search, o, $t^{\mathrm{it}}_{\mathrm{RF}}$ and $t^{\mathrm{it}}_{\mathrm{FO}}$ are increased together. At this point, the tests are stopped even if no feasible solution is found.

The number of intervals belonging to the end block when starting the R&F algorithm was set to $p - 2$ in all tests, which is the maximum value allowed for the algorithm. The initial optimality gap was set to 50%. The F&O strategies used were IV and PT (see Section 4.4). For the first strategy, which divides the planning horizon into m intervals, the parameter m was set to 3. For all F&O strategies, the optimality gap was set to 0.1%. The time limit $t^{\max}_{\mathrm{FO}}$ of the improvement phase is equally divided between the local searches used in each test. If some local search finishes before reaching the time limit, the remaining time is available for the next local search.

The chosen parameters and results of the time-space network model are presented in Table 5.4. Column *Parameters* presents the parameters' values, columns *R&F* presents the results considering only the relax-and-fix algorithm, while columns *F&O* present the results concerning the fix-and-optimize algorithm performed over the solution obtained by the R&F. Column *Obj* corresponds to the objective value, and column *Time* corresponds to each algorithms' processing time (in seconds). Column *Gap*^{LB} presents the gap deviation of the objective function value Obj in relation to the lower bound LB obtained by the TS model linear relaxation with the additional constraints described in Section 3.5, being calculated as $(\frac{Obj-LB}{-LB}) \cdot 100$. When the row is singed with a "-", no feasible solution was found for the corresponding instance. Finally, column *Total Time* presents the total time spent by the algorithm.

According to Table 5.4, The F&O algorithm improved the objective function on average 11.4%, considering only the cases where R&F provided a feasible solution, i.e., without using the auxiliary variables β_{it}. Considering the instance where R&F has not provided a feasible solution, F&O could remove the infeasibility in five instances. On the other hand, on average, 85.4% of the total time was spent in the improvement phase. The algorithms could not obtain even a feasible solution for eight instances, six with time horizon $T = 60$.

Table 5.4: Results for the time-space model with MIRPLIB instances

Instance	Parameters					R&F		F&O			Total
	$\beta_{i,t}$	p	o	$t^{\text{it}}_{\text{RF}}$	$t^{\text{it}}_{\text{FO}}$	Time	Obj	Time	Obj	Gap$^{\text{LB}}$	Time
$T = 45$											
LR1_1_DR1_3_VC1_V7a		5	15	50	25	55	-13,178	454	-13,272	23.3%	509
LR1_1_DR1_4_VC3_V11a		5	15	50	25	159	-10,682	822	-10,910	35.0%	981
LR1_1_DR1_4_VC3_V12a		5	15	50	25	38	-8,540	624	-10,372	35.0%	662
LR1_1_DR1_4_VC3_V12b		5	15	50	25	179	-7,999	1,611	-9,057	46.5%	1,790
LR1_1_DR1_4_VC3_V8a		9	30	50	25	26	-4,688	86	-5,106	31.7%	112
LR1_1_DR1_4_VC3_V9a		5	15	50	25	29	-5,419	649	-6,629	36.2%	678
LR1_2_DR1_3_VC2_V6a		5	30	50	25	256	-9,511	1,797	-10,577	45.8%	2,053
LR1_2_DR1_3_VC3_V8a		5	15	50	25	188	-10,133	1,255	-11,680	39.5%	1,443
LR2_11_DR2_22_VC3_V6a	•	5	50	400	300	1,610	94,720	5,401	-9,550	50.0%	7,011
LR2_11_DR2_33_VC4_V11a	•	5	15	100	75	395	565,310	5,475	-13,218	62.5%	5,870
LR2_11_DR2_33_VC5_V12a	•	5	15	100	75	435	747,441	5,551	-15,125	64.1%	5,986
LR2_22_DR2_22_VC3_V10a	•	5	50	400	300	2,035	1,374,400	5,436	-21,957	43.6%	7,471
LR2_22_DR3_333_VC4_V14a	•	5	15	100	75	585	1,825,910	6,183	25,843	-	6,768
LR2_22_DR3_333_VC4_V17a	•	5	50	400	300	2,392	6,723,570	9,047	2,033	-	11,439
Average						599		3,171		42.8%	3,769
$T = 60$											
LR1 1 DR1 3 VC1 V7a		6	50	50	25	176	-16,326	776	-16,675	18.9%	952
LR1_1_DR1_4_VC3_v11a		6	15	50	25	214	-11,113	2,312	-11,516	40.0%	2,526
LR1_1_DR1_4_VC3_V12a	•	6	15	50	25	190	-10,012	1,584	-11,223	35.2%	1,773
LR1_1_DR1_4_VC3_V12b		6	15	100	25	481	-8,018	2,960	-9,958	44.7%	3,441
LR1_1_DR1_4_VC3_V8a	•	6	15	50	35	115	325,680	865	-4,578	34.9%	980
LR1_1_DR1_4_VC3_V9a		6	15	50	25	178	-6,746	757	-6,904	38.6%	935
LR1_2_DR1_3_VC2_V6a		10	15	30	25	172	-10,514	1,869	-12,639	47.9%	2,040
LR1_2_DR1_3_VC3_V8a		6	15	200	25	495	-12,857	1,573	-14,329	42.8%	2,068
LR2_11_DR2_22_VC3_V6a	•	6	15	100	75	245	195,984	5,403	39,102	-	5,648
LR2_11_DR2_33_VC4_V11a	•	6	15	100	75	560	1,523,420	5,702	181,868	-	6,262
LR2_11_DR2_33_VC5_V12a	•	6	15	100	75	601	906,791	5,702	176,226	-	6,304
LR2_22_DR2_22_VC3_V10a	•	6	15	100	75	639	2,685,950	5,823	826,372	-	6,462
LR2_22_DR3_333_VC4_V14a	•	6	15	100	75	1,094	7,925,830	6,222	2,218,890	-	7,316
LR2_22_DR3_333_VC4_V17a	•	6	30	200	150	2,134	4,228,370	5,732	1,110,680	-	7,866
Average						521		3,377		37.9%	3,898

Source: From the author (2020).

One can observe that for the smaller instances (those with one loading region - LR1), the algorithms performed better, using smaller values for the parameters, which contributed to the short processing times (mainly for the R&F). Another aspect observed is that in most cases, it was not necessary the use auxiliary variables β_{it}, demonstrating that the R&F can obtain easily feasible solutions for such instances. If it is preferable to obtain a better objective value than a short processing time for the small instances, the parameters' values can be increased (except p).

For the larger instances (those with two loading regions - LR2), we can observe that, as expected, it was necessary to increase the parameter values for obtaining the solutions. As a consequence, the processing time of the F&O increased considerably. It was also more difficult for the R&F to obtain feasible solutions, as even the use of β_{it} variables is not sufficient to guarantee it. Preliminary tests obtained new best-known values for two instances with $T = 45$, presented in Table 5.5. Column *CPU* presents the computer where the experiments were carried out, where "AMD" corresponds to the previously described computer, while “Core i5” corresponds to an Intel Core i5-2300 running at 2.8 GHz, with 16 GB. Both experiments used the F&O strategies IV and PT described in Section 4.4. Also, they did not use auxiliary variables β_{it}. Column *Gap*^{BKV} presents the relative deviation $(\frac{Obj-BKV}{-BKV}) \cdot 100$ from the best-known value, where BKV corresponds to the objective value found by Papageorgiou et al. (2014a).

Table 5.5: New best-known values found in preliminary experiments with the TS model

Instance	Parameters					R&F		F&O		BKV^{LB}	Gap^{BKV}
	CPU	p	o	t^{it}_{RF}	t^{it}_{FO}	Time	Obj	Time	Obj		
LR1_1_DR1_4_VC3_V11a	Core i5	5	20	50	50	161	-10,448	1,578	-11,243	33.1%	-0.03%
LR1_1_DR1_4_VC3_V12b	AMD	5	15	50	20	182	-6,606	1,942	-9,085	46.4%	-0.17%

Source: From the author (2020).

From the new best-known values found, the improvement on the solution by the R&F was on average 22.6%. Notice that Gap^{LB} is higher for the solutions obtained, as the linear relaxation of the model is weak.

5.3.1.1 Modified Instances Results

For solving the modified instances, the same parameters and methodology of the tests with the MIRPLIB instances were used. The parameters used for each instance and the obtained results are presented in Table 5.6.

Considering the results presented in Table 5.6, the relative gap to the lower bounds was, on average smaller than the gap in the tests with MIRPLIB instances. This information does not necessarily mean that the proposed algorithm is better considering these instances, as the model’s linear relaxation can be tighter in randomly distributed ports. The solution approach also found feasible solutions for a significant number of instances compared to the number of feasible solutions found for the MIRPLIB instances. However, there are still instances that no feasible solution was found. The average improvement of the objective function with the improvement phase was 9.7%, while the time spent in this phase was, on average, 87.6%.

5.3.2 FCNF+ Model Results

The results presented in this section were obtained using the CPLEX 12.5 solver, running in a Core i7-3632QM computer running at 2.2 GHz on a single core, with 8 GB RAM.

The parameters for solving the FCNF+ model with R&F and F&O are different

Table 5.6: Results for the TS model with the modified instances.

Instance	Parameters					Relax and Fix		Fix and Optimize			Total
	β_{it}	p	o	$t^{\text{it}}_{\text{rf}}$	$t^{\text{it}}_{\text{ls}}$	Time	Obj	Time	Obj	Gap^{LB}	Time
$T = 45$											
LR1_1_DR1_3_VC1_V7a		5	15	50	25	12	-19,547	986	-21,491	5.5%	999
LR1_1_DR1_4_VC3_V11a		5	15	50	25	65	-22,164	2,556	-24,617	8.0%	2,621
LR1_1_DR1_4_VC3_V12a		9	30	50	25	136	-22,263	1,618	-23,062	6.1%	1,754
LR1_1_DR1_4_VC3_V12b		5	15	50	25	129	-23,883	3,625	-26,842	5.9%	3,754
LR1_1_DR1_4_VC3_V8a		5	15	50	25	79	-15,731	2,797	-17,342	13.0%	2,876
LR1_1_DR1_4_VC3_V9a		5	15	100	25	123	-16,196	2,101	-17,324	14.1%	2,224
LR1_2_DR1_3_VC2_V6a		5	15	50	25	43	-17,896	1,451	-19,597	8.8%	1,494
LR1_2_DR1_3_VC3_V8a		5	15	100	25	125	-19,138	1,827	-20,568	14.8%	1,952
LR2_11_DR2_22_VC3_V6a		5	15	100	50	171	-14,017	5,410	-15,064	27.2%	5,581
LR2_11_DR2_33_VC4_V11a		9	15	30	50	144	-24,328	5,450	-27,728	21.6%	5,594
LR2_11_DR2_33_VC5_V12a	•	5	15	100	75	441	925,516	5,852	-22,581	46.4%	6,293
LR2_22_DR2_22_VC3_V10a	•	5	50	400	300	2,041	1,397,820	6,002	-23,326	33.9%	8,043
LR2_22_DR3_333_VC4_V14a	•	5	15	100	75	597	891,797	5,706	-28,232	41.4%	6,303
LR2_22_DR3_333_VC4_V17a	•	5	15	100	75	756	8,208,870	5,941	62,221	-	6,697
Average						347		3,666		19.0%	4,013
$T = 60$											
LR1 1 DR1 3 VC1 V7a		6	15	50	25	69	-24,995	1,584	-27,275	6.2%	1,653
LR1_1_DR1_4_VC3_v11a		6	15	50	25	148	-26,952	3,625	-31,455	7.9%	3,774
LR1_1_DR1_4_VC3_V12a	•	6	30	50	25	198	-27,495	3,564	-29,613	2.9%	3,762
LR1_1_DR1_4_VC3_V12b		6	15	50	25	165	-33,163	3,625	-34,264	6.4%	3,790
LR1_1_DR1_4_VC3_V8a		6	15	100	25	214	-19,355	2,552	-20,905	17.9%	2,765
LR1_1_DR1_4_VC3_V9a		6	15	100	25	172	-18,922	3,701	-21,640	17.0%	3,873
LR1_2_DR1_3_VC2_V6a		6	15	50	25	165	-22,908	2,377	-25,324	9.8%	2,542
LR1_2_DR1_3_VC3_V8a		6	15	100	25	277	-23,370	1,827	-25,687	17.3%	2,104
LR2_11_DR2_22_VC3_V6a	•	6	15	100	75	432	625,009	5,701	-19,230	27.7%	6,133
LR2_11_DR2_33_VC4_V11a	•	6	15	100	75	660	504,184	5,551	-32,807	29.5%	6,212
LR2_11_DR2_33_VC5_V12a	•	6	50	400	300	2,554	2,632,490	7,204	173,385	-	9,758
LR2_22_DR2_22_VC3_V10a	•	6	15	100	50	604	1,514,310	5,787	92,707	-	6,392
LR2_22_DR3_333_VC4_V14a	•	6	30	200	150	2,030	2,728,360	5,863	517,543	-	7,893
LR2_22_DR3_333_VC4_V17a	•	6	30	200	150	2,526	8,152,140	6,653	3,014,670	-	9,179
Average						730		4,258		14.3%	4,988

Source: From the author (2020).

from those used for the TS model. To provide more robustness to the proposed solution approach, we defined at most two possible values for each parameter, independently of the instance and the planning horizon's size. Table 5.7 presents the parameters values tested for the FCNF+ model.

The improvement phase used the F&O strategies described in Section 4.4 in the following order: IV, VP, and TI. As in the TS model, the improvement phase's total time limit was divided equally for each strategy. The MIP relative gap was set as the default value of the CPLEX solver in the improvement phase.

For evaluating the contribution of the valid inequalities in the matheuristic approach, Table 5.8 presents the computational results using the FCNF+ model without and with the use of valid inequalities. Column Gap^{LB} presents the relative deviation gap of the integer solution to the linear relaxation of the FCNF model with the valid inequalities, and column Gap^{BKV}presents the relative deviation from the best-known value found by Papageorgiou et al. (2014a). This value is negative when a new best-known value was

Table 5.7: Parameters and values used for the FCNF model

	Parameter	Value	
R&F	Number of time periods per interval (T/p)	15	
	Size of model (in time periods) at the first iteration	45	Instances prefix “LR1_1”
		30	Instances prefix “LR1_2” or “LR2”
	Overlap (%)	50	
	Time limit (s) per iteration	600	Instances prefix “LR1” or “LR2_11”
		1200	Instances prefix “LR2_22”
	Initial MIP GAP (%)	5	
F&O	Time limit (s) per iteration	120	
	Total time limit (s)	7200	
	Number of time intervals m	3	

Source: from the author (2020).

found by the proposed method. The best results concerning the value of the objective function are highlighted in bold.

Table 5.8: Results of FCNF+ model and FCNF+ model with valid inequalities.

Instance	FCNF+							FCNF+ with valid inequalities						
	Relax and fix		Fix and optimize		Total time	Gap^{LB}	Gap^{BKV}	Relax and fix		Fix and optimize		Total time	Gap^{LB}	Gap^{BKV}
	Time	Obj	Time	Obj				Time	Obj	Time	Obj			
$T = 45$														
LR1_1_DR1_3_VC1_V7a	88	-13,155	2,597	-13,191	2,685	7.5%	0.6%	101	-13,271	375	**-13,272**	476	6.9%	0.0%
LR1_1_DR1_4_VC3_V11a	949	-10,961	4,377	**-11,094**	5,326	5.0%	1.3%	912	-10,537	1,879	-11,009	2,792	5.7%	2.1%
LR1_1_DR1_4_VC3_V12a	1,366	-10,578	2,854	**-10,715**	4,220	4.1%	0.2%	1,421	-10,492	2,721	-10,709	4,142	4.2%	0.2%
LR1_1_DR1_4_VC3_V12b	799	-9,057	382	**-9,057**	1,182	5.3%	0.3%	1,234	-9,023	2,469	-9,028	3,703	5.6%	0.6%
LR1_1_DR1_4_VC3_V8a	795	-5,019	2,095	**-5,071**	2,890	7.0%	0.7%	296	-5,060	2,039	-5,060	2,335	7.2%	0.9%
LR1_1_DR1_4_VC3_V9a	767	-6,904	2,632	**-6,921**	3,399	4.5%	-0.4%	728	-6,921	492	**-6,921**	1,220	4.5%	-0.4%
LR1_2_DR1_3_VC2_V6a	88	-10,965	1,280	**-10,966**	1,368	4.6%	1.5%	729	-10,455	1,652	-10,717	2,380	6.8%	3.7%
LR1_2_DR1_3_VC3_V8a	252	-11,587	2,966	**-11,938**	3,218	1.1%	0.6%	296	-10,658	906	-11,889	1,202	1.5%	1.0%
LR2_11_DR2_22_VC2_V6a	1,304	-8,720	2,322	**-9,620**	3,626	20.5%	1.0%	225	149,598	1,775	-8,510	2,000	29.7%	12.4%
LR2_11_DR2_33_VC4_V11a	2,054	29,467	3,330	**-13,340**	5,384	14.0%	4.8%	2,465	22,761	2,875	-11,651	5,340	24.9%	16.9%
LR2_11_DR2_33_VC5_V12a	1,694	-17,900	4,613	-18,208	6,307	6.9%	1.2%	1,418	-18,083	3,232	**-18,395**	4,650	5.9%	0.2%
LR2_22_DR2_22_VC3_V10a	3,498	-23,810	5,041	-24,784	8,539	10.7%	0.0%	1,742	-23,905	4,148	**-24,855**	5,890	10.5%	-0.3%
LR2_22_DR3_333_VC4_V14a	5,845	15,699	4,922	**-22,005**	10,767	13.2%	-0.2%	5,242	-20,013	5,043	-21,925	10,285	13.5%	0.1%
LR2_22_DR3_333_VC4_V17a	6,565	-4,038	5,239	-21,569	11,804	14.3%	0.7%	5,748	-21,253	5,045	**-22,294**	10,793	11.4%	-2.7%
AVG	1,862		3,189		5,051	8.5%	0.9%	1,611		2,475		4,086	9.9%	2.5%
$T = 60$														
LR1_1_DR1_3_VC1_V7a	799	-16,328	4,843	-16,522	5,642	6.2%	0.9%	732	-16,673	2,190	**-16,675**	2,922	5.4%	0.0%
LR1_1_DR1_4_VC3_V11a	2,009	-12,126	5,042	-12,134	7,051	11.0%	8.5%	2,211	-11,249	4,806	**-12,155**	7,018	10.9%	8.3%
LR1_1_DR1_4_VC3_V12a	1,359	-10,870	2,942	**-11,125**	4,301	4.1%	-0.8%	1,348	-10,907	4,419	-11,022	5,768	5.0%	0.2%
LR1_1_DR1_4_VC3_V12b	1,216	-9,883	4,580	**-10,009**	5,796	4.6%	0.4%	2,122	-9,785	4,735	-9,830	6,857	6.3%	2.2%
LR1_1_DR1_4_VC3_V8a	724	-4,372	4,469	**-5,182**	5,193	9.7%	0.2%	1,684	-4,410	2,829	-4,502	4,513	21.6%	13.3%
LR1_1_DR1_4_VC3_V9a	1,848	-7,288	3,484	**-7,335**	5,332	8.8%	2.9%	2,032	-7,278	4,494	-7,328	6,526	8.8%	3.0%
LR1_2_DR1_3_VC2_V6a	1,566	-12,533	4,921	-12,807	6,487	10.1%	6.0%	1,192	-12,406	3,162	**-12,843**	4,354	9.9%	5.8%
LR1_2_DR1_3_VC3_V8a	1,840	-12,199	5,043	-13,945	6,883	7.8%	4.8%	1,114	-13,817	3,752	**-14,152**	4,866	6.4%	3.4%
LR2_11_DR2_22_VC2_V6a	1,866	175,037	4,230	15,523	6,097	-	-	420	223,498	4,373	42,126	4,793	-	-
LR2_11_DR2_33_VC4_V11a	3,820	79,086	5,164	**-15,865**	8,984	14.0%	-3.1%	3,552	39,783	5,043	-14,379	8,595	22.1%	6.6%
LR2_11_DR2_33_VC5_V12a	2,492	-20,171	5,043	-20,717	7,536	15.2%	8.9%	2,453	59,566	3,882	**-22,948**	6,335	6.1%	-1.0%
LR2_22_DR2_22_VC3_V10a	3,812	-31,862	4,997	**-32,529**	8,809	7.7%	0.3%	4,143	-5,356	5,043	-31,598	9,186	10.4%	3.2%
LR2_22_DR3_333_VC4_V14a	8,687	170,058	5,589	20,479	14,277	-	-	7,669	-23,307	5,047	**-25,069**	12,716	18.6%	6.7%
LR2_22_DR3_333_VC4_V17a	-	-	-	-	-	-	-	9,558	6,874	5,461	**-25,236**	15,020	18.7%	6.5%
AVG	2,465		4,642		7,107	9.0%	2.6%	2,874		4,231		7,105	11.5%	4.5%

Source: from the author (2020).

One can observe, according to Table 5.8, that the use of valid inequalities speed-up the solution approach on average. For $T = 45$, the time required for solving the instances when using the valid inequalities was, on average, 23% lower than without using them. For $T = 60$, this value was 9.4%, disregarding instance "LR2_22_DR3_333_VC4_V17a", in which the time for solving each interval (1200 s) was not sufficient to obtain even an integer solution during some iteration of the relax and fix. Without valid inequalities, the model provided better solutions on average, considering the deviation gap from the best-known values. However, there are three instances in which no feasible solution was found using only the FCNF+ model, while when considering valid inequalities, there is only one instance. The number of new best-known values found in the model with and without valid inequalities was the same (4). As both tests found new BKVs for instance "LR1_1_DR1_4_VC3_V9a" with $T = 45$, a total of seven new BKVs was found.

In the FCNF+ model with valid inequalities, the improvement phase consumed, on average, 62% of the total processing time and removed the infeasibility in the solution provided by the R&F for six instances. When no valid inequalities were considered, the F&O consumed on average 64% of the total time, and the infeasibility was removed from four instances. Excluding the cases where the R&F provided infeasible solutions, the average improvement on the solution was approximately 3.1%. For some instances no improvement was obtained by the F&O algorithm.

Previous tests were performed using a small number of time periods per interval for the R&F (9 time periods for instances with $T = 45$ and 10 time periods for instances with $T = 60$). Although not presented here, these results showed that the approach can be faster for small instances and still provide good solutions, sometimes equal to the BKV. However, for large instances, the approach becomes very myopic, and the solution quality decreases considerably. This phenomenon was already observed by Stadtler (2003) for solving the multi-stage lot-sizing problem. For comparison purposes with the TS model results of Section 5.3.1, the results of FCNF+ with valid inequalities will be used.

5.3.3 Comparison with the Best-known Values

This section presents a comparative between the results of the TS model (Section 5.3.1) and FCNF+ model (Section 5.3.2) using the matheuristic framework proposed. The results were also compared with the work of Papageorgiou et al. (2014a), which holds the best-known values for the tested MIRPLIB instances. For providing results more

Table 5.9: Time coefficients values for normalizing the CPU processing times.

Processor	Time coefficient
Intel Core i7-3632QM (Reference - FCNF+ model tests)	1.00
Intel Xeon E5520[Dual] (PAPAGEORGIOU et al., 2014a)	1.07
Intel Core i5-2300 (TS model tests)	0.77
AMD-FX-8150 (TS model tests)	0.91

Source: from the author (2020).

comparable, the CPU times were normalized using the *PassMark Software*[2]. Table 5.9 shows the time normalization coefficients for each processor.

Table 5.10 presents the final objective value and the total computational time of each approach. Column *Time-space* refers to the results presented in Section 5.3.1, while column *FCNF+* corresponds to the results presented in Section 5.3.2, and column *BKV* corresponds to the results of Papageorgiou et al. (2014a). For the results of the TS model, Table 5.10 presents the two best-known values shown in Table 5.5. Column *Gap* is related to the relative deviation $(\frac{Obj-BKV}{-BKV}) \cdot 100$ from the best-known value BKV. The best objective value for each instance is highlighted in bold.

According to Table 5.10, the matheuristic framework applied to the FCNF model performed on average better than the time-space model considering solution quality and the number of instances in which a feasible solution was found. Considering the average processing time, the difference between the models for $T = 45$ is almost negligible, while for $T = 60$, the FCNF model is more time-consuming. One possible explanation is that the FCNF+ model has a larger number of constraints and variables, taking more time for solving each MIP. However, this is not a rule as it can be observed that less processing time is needed for the FCNF model in some cases.

Comparing the proposed approaches with the results of Papageorgiou et al. (2014a), one can note that for both formulations, the matheuristic framework takes on average less processing time. Also, new best-known values were found for a total of nine instances: two when using the TS model and seven when using the FCNF+ model, for which three BKVs were obtained without the use of valid inequalities. Except for three instances in which the deviation gap from the BKV was greater than 10%, the average gap deviation was 0.5% for $T = 45$ and 3.73% for $T = 60$ using the FCNF+ model, which can be considered very satisfactory as the proposed framework is relatively simple if compared with the two-stage algorithm proposed by Papageorgiou et al. (2014a). Such observations demonstrate no dominance between the proposed solution approach and the method of

[2]<http://www.cpubenchmark.net/>

Table 5.10: Comparison between TS and FCNF+ model considering the matheuristic framework.

Instance	TS			FCNF+			BKV	
	Time	Obj	Gap	Time	Obj	Gap	Time	Obj
$T = 45$								
LR1_1_DR1_3_VC1_V7a	559	**-13,272**	0.0%	476	**-13,272**	0.0%	190	**-13,272**
LR1_1_DR1_4_VC3_V11a	1,436	**-11,243**	0.0%	2,792	-11,009	2.0%	9,151	-11,239
LR1_1_DR1_4_VC3_V12a	727	-10,372	3.4%	4,142	-10,709	0.2%	8,566	**-10,732**
LR1_1_DR1_4_VC3_V12b	1,768	**-9,085**	-0.2%	3,703	-9,028	0.4%	1,906	-9,069
LR1_1_DR1_4_VC3_V8a	123	**-5,106**	0.0%	2,335	-5,060	0.9%	4,943	**-5,106**
LR1_1_DR1_4_VC3_V9a	744	-6,629	3.8%	1,220	**-6,921**	-0.4%	549	-6,891
LR1_2_DR1_3_VC2_V6a	2,255	-10,577	5.0%	2,380	-10,717	3.7%	6,396	**-11,134**
LR1_2_DR1_3_VC3_V8a	1,585	-11,680	2.8%	1,202	-11,889	1.0%	8,188	**-12,010**
LR2_11_DR2_22_VC2_V6a	7,702	-9,550	1.7%	2,000	-8,510	12.4%	8,740	**-9,718**
LR2_11_DR2_33_VC4_V11a	6,449	-13,218	5.7%	5,340	-11,651	16.9%	9,559	**-14,017**
LR2_11_DR2_33_VC5_V12a	6,576	-15,125	17.9%	4,650	-18,395	0.2%	9,582	**-18,423**
LR2_22_DR2_22_VC3_V10a	8,208	-21,957	11.4%	5,890	**-24,855**	-0.3%	9,359	-24,789
LR2_22_DR3_333_VC4_V14a	7,436	25,843	-	10,285	-21,925	0.1%	10,088	**-21,952**
LR2_22_DR3_333_VC4_V17a	12,567	2,033	-	10,793	**-22,294**	-2.7%	10,218	-21,713
AVG	4,153		4.3%	4,086		2.5%	6,399	
$T = 60$								
LR1_1_DR1_3_VC1_V7a	1,046	**-16,675**	0.0%	2,922	**-16,675**	0.0%	476	**-16,675**
LR1_1_DR1_4_VC3_V11a	2,775	-11,516	13.1%	7,018	-12,155	8.3%	7,657	**-13,257**
LR1_1_DR1_4_VC3_V12a	1,948	-11,223	0.4%	5,768	-11,022	0.2%	8,566	**-11,269**
LR1_1_DR1_4_VC3_V12b	3,780	-9,958	0.9%	6,857	-9,830	2.2%	9,342	**-10,053**
LR1_1_DR1_4_VC3_V8a	1,077	-4,578	11.8%	4,513	-4,502	13.3%	8,245	**-5,191**
LR1_1_DR1_4_VC3_V9a	1,027	-6,904	8.6%	6,526	-7,328	3.0%	8,886	**-7,552**
LR1_2_DR1_3_VC2_V6a	2,241	-12,639	7.3%	4,354	-12,843	5.8%	8,902	**-13,631**
LR1_2_DR1_3_VC3_V8a	2,272	-14,329	4.0%	4,866	-14,152	3.4%	8,613	**-14,931**
LR2_11_DR2_22_VC2_V6a	6,205	39,102	-	4,793	42,126	-	9,014	**-13,351**
LR2_11_DR2_33_VC4_V11a	6,880	181,868	-	8,595	-14,379	6.6%	9,592	**-17,008**
LR2_11_DR2_33_VC5_V12a	6,925	176,226	-	6,335	**-22,948**	-1.0%	9,656	-22,730
LR2_22_DR2_22_VC3_V10a	7,100	826,372	-	9,186	-31,598	3.2%	9,441	**-32,627**
LR2_22_DR3_333_VC4_V14a	8,038	2,218,890	-	12,716	-25,069	6.7%	10,234	-26,873
LR2_22_DR3_333_VC4_V17a	8,642	1,110,680	-	15,020	-25,236	6.5%	10,312	-27,000
AVG	4,283		5.8%	7,105		4.5%	8,495	

Source: from the author (2020).

Papageorgiou et al. (2014a) considering solution quality or processing time. For example, for small instances, the R&F using the TS model can be the preferable approach to obtain good solutions in a short processing time. If the objective is to achieve better solutions, the R&F with the FCNF+ is preferable in some cases. If the objective is to obtain a feasible solution for the instances, independently of their size, then it is preferable to use the method of Papageorgiou et al. (2014a).

5.3.4 Parameter Calibration

The tests presented in this section were obtained using the CPLEX 12.9 solver and carried out on an AMD Ryzen 7 3700X computer running at 2.2 GHz on a single core, with 16 GB RAM. Different combinations involving the formulation components,

Table 5.11: Formulation components and algorithms parameters evaluated by *irace*.

	Parameter/Component	Evaluated Values	Chosen Values		
			TS	FCNF	
Formulation	Operate-and-depart constraints. (Eq. (3.65))	On/Off	Off	Off	On
	Valid inequalities (Sect. 3.4)	On/Off	Off	On	On
	Tight flow variables (Sect. 3.2.1)	On/Off	-	On	Off
R&F	Size of each interval (in time periods)	{5,10,15,20}	10	5	20
	End-block initial size (in intervals)	{0,1,2,3,4,5,6,7,8,9,10}	4	3	1
	Proportional $\alpha_i^{\max}$ (Sect. 4.3.3)*	On/Off	Off	Off	On
	Tight Inventory (Sect. 4.3.3)*	On/Off	On	On	On
	Initial gap (%)	{0.01, 5, 25, 50}	5	0.01	0.01
	Overlap (%)	{0, 10, 25, 50}	50	50	10
F&O	Strategies and order	All permutations	{PT,TI,IV,VP}	{VP,IV,PT,TI}	{VP,IV,PT}
	Number of time intervals (TI)	{2,3}	2	2	2
	Number of time intervals (IV)	{2,3}	3	2	-
	Overlap u^{TI} between intervals (TI)	{0,25,50}	50	25	50
	Overlap u^{VTI} between intervals (IV)	{0,25,50}	50	50	-

Source: From the author (2020).

the R&F and F&O algorithms' parameter values, and the F&O strategies' order of execution can be used to solve the problem. Trying to define the best one to obtain high-quality solutions with regards to the objective function value instead of using a priori parameter setting (sections 5.3.1-5.3.2), we used the *irace* package (LÓPEZ-IBÁÑEZ et al., 2016), an automatic algorithm configuration tool. We performed separated parameter calibrations for TS and FCNF model. As training instances, we have chosen four instances ($< 15\%$ of the total number of tested instances) considering different characteristics and sizes. The instances were:

- LR1_2_DR1_3_VC3_V8a and LR2_22_DR3_333_VC4_V14a, for $T = 45$,
- LR1_1_DR1_3_VC1_V7a and LR2_11_DR2_33_VC4_V11a, for $T = 60$.

We limited our algorithm's maximum total execution time to 1,728,000 seconds (20 days) for each calibration. Considering that *irace* can test different configurations in parallel (we used up to 14 cores), the calibration time was shorter, taking on average six days. We set a fixed time limit of 600 and 1000 seconds to solve each R&F subproblem for the instances with one loading region (LR1) and two loading regions (LR2), respectively. The time limit for solving each F&O iteration and the total time for each strategy was set to 120 seconds and 1 hour, respectively.

Table 5.11 presents the formulation components, the R&F and F&O parameters, and their evaluated values for the TS and FCNF models. Columns *Selected Values* present the best parameter values defined by the best-ranked configuration of *irace*. We present two configurations for the FCNF model: the first column presents the best-ranked configuration by the *irace*, while the second presents the parameter values used in the experiments.

Observe from Table 5.11 that not all combinations between R&F parameters are compatible, depending on the instance. For example, if the interval size was set to 5 time periods, for an instance of $T = 45$, then the end block's maximum size is 7, not 10. Such forbidden combinations are handled directly by *irace*. From the results presented in Table 5.11, we remark that:

- A small value of the initial gap(%) was selected for the formulations. The value 0.01 means that the gap is the same for all iterations (as the value at the last iteration is always 0.01 - see Section 4.3.1). This configuration implies in consuming more processing time in the R&F algorithm.

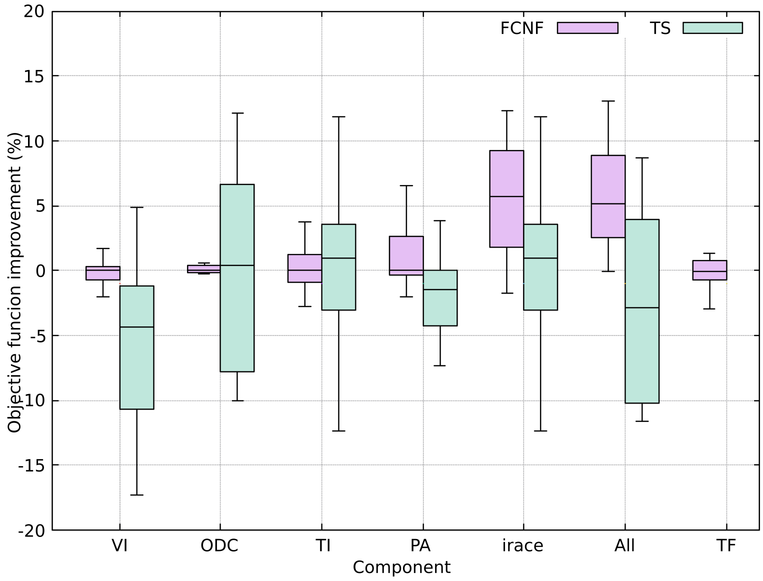
- The maximum value of overlap was also selected, indicating that the highest the overlap, the highest the solution quality. Such characteristic was already stated by Uggen, Fodstad and Nørstebø (2013).

- An end block with a size greater than 0 is considered in both formulations and the tight inventory is also adopted, while the proportional α_i^{max} does not seem to be effective.

- In two cases, all F&O strategies were selected, but the order differs depending on the formulation.

Due to the previous experiments using the R&F, we have observed that time intervals with small sizes (≤ 10) combined with an end block size > 0 tend to decrease the algorithm's processing time. On the other hand, considering the FCNF model only, the probability of not finding a feasible solution for some instances increases due to the reasons reported in Section 4.3.2. This was the case of the first configuration of FCNF model shown in column *Selected values* in Table 5.11. As the *irace* presents five best parameter calibrations that statistically have no difference considering the solution quality, we decided to use another configuration given by *irace* in which the size of each time period is larger than the one reported by the best-ranked configuration. This configuration is shown in the last column of Table 5.11. Note that the overlap parameter of the R&F in such configuration is 10%. Considering that the other four configurations given by *irace* defined the overlap value to 50%, and preliminary tests demonstrated that an overlap of 10% increased the number of infeasible solutions considerably, we decided to adjust the overlap value to 50%.

Figure 5.1: Objective function's improvement obtained by the formulation components in comparison to the default case.



Source: From the author (2020).

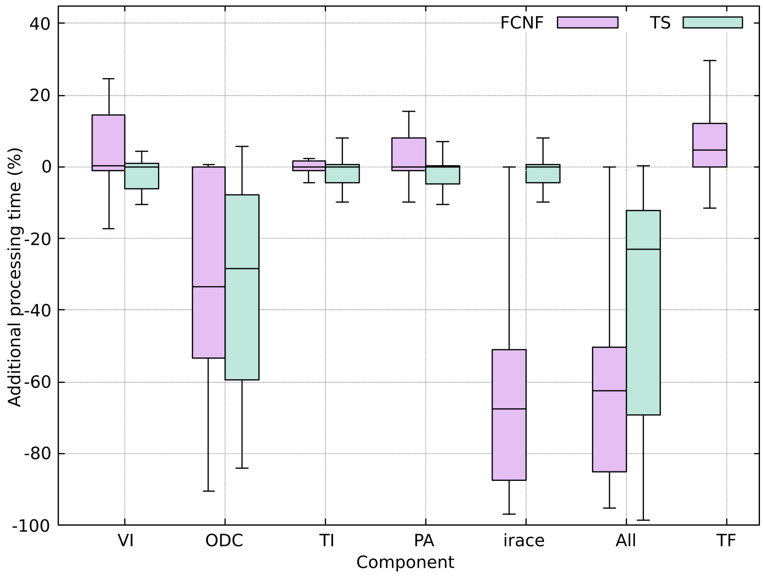
5.3.5 Evaluating Formulation Components and R&F Features

We analyzed how much the formulation components and the end block features contributed to the solution quality and affected the R&F processing time. We first run the algorithm with the parameters defined in Table 5.11 with all components and features disabled (*default case*). Then, several runs of the R&F were performed, each one with only one formulation component activated. We also performed a run with all formulation components activated. The boxplot in Figure 5.1 presents the improvement of the default case objective function by each component, where *VI* = valid inequalities, *ODC* = operate-and-depart constraints, *TI* = Tight Inventory, *PA* = Proportional $\alpha_i^{\max}$, *irace* = irace configuration, *TF* = tight flow variables (only for FCFN model). The improvement was calculated only for the instances in which a feasible solution was obtained by the default case and by the configuration with the component/feature activated.

From Figure 5.1, we observe that there are negative improvements in most cases, i.e., the solution quality, for some instances, decreases if one or more components are considered. Also, the improvement usually varies more for the TS model than for the FCNF model, depending on the instance. The *irace* configuration and the case with all components activated provided the best results for the FCNF model, where the median improvement was above 5% (indicated by the horizontal line in the middle of the box), indicating that it occurs for 50% of the instances. The same cannot be stated for the TS model since the third quartile (superior limit of the box) indicate that 75% of the instance obtained an improvement below to 4% approximately, besides larger negative improvements. A possible reason is that the processing time of R&F using the TS model is, on average, 94% greater than the FCNF model (considering the default case). Adding components such as the valid inequalities increase the processing time, and possibly the iterations stop prematurely (not by reaching the stipulated gap), not necessarily showing the component's real improvement.

Figure 5.2 presents the additional processing time required by the components.

Figure 5.2: Increase in the R&F processing time considering different formulation components.



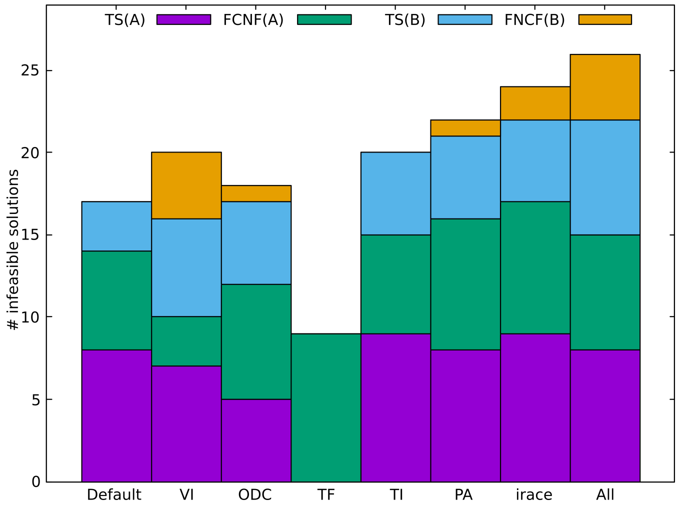
Source: From the author (2020).

From Figure 5.2, we observe that the operate-and-depart constraints contributed significantly to reducing the processing time of the R&F in both formulations. Combining all components and using the *irace* configuration (only for the FCNF model) induces the fastest processing times for the R&F. In contrast, for the other components, the time difference is negligible on average.

Figure 5.3 presents the number of instances with infeasible solutions due to the use of auxiliary variables β_{it}, θ_{it} (A), and due to a premature stop of the R&F algorithm (explained in Section 4.3.2) or no feasible solution found in the stipulated processing time (B).

Figure 5.3 contrasts with the results shown in Figure 5.1. While using all components and the *irace* configuration provided in general better objective function values

Figure 5.3: Number of instances with infeasible solutions using the R&F considering the TS and FCNF models.



Source: From the author (2020).

and less processing times, they also provided the largest number of infeasible solutions of type B.

The results on Figures 5.1–5.3 demonstrate that it is hard to define only one parameter configuration for obtaining good results for all tested instances and in all cases of the R&F. It is necessary to identify instance features that can be used to choose the best algorithm configuration for each instance. This can be done using algorithm selection tools based on machine learning methods.

5.3.6 Fix-and-optimize Algorithm Results

We evaluated the improvement of the solution obtained by each F&O strategy over the R&F solution, besides the capacity to recover the solutions that used the additional variables β_{it} and θ_{it}. For the tests, we considered the parameters defined by the *irace* for both R&F and F&O. Table 5.12 presents the minimum, maximum, and average values considering improvement percentage and time consumption in seconds for each F&O strategy individually. It also presents the results from the *irace* configuration, considering more than one strategy and its order of execution. The improvement was measured only for the instances where R&F obtained a feasible solution. Column *Infeasibility Removed* presents the number of instances in which the F&O strategy could repair the solution, turning it feasible. The total number of infeasible solutions given by the R&F was 9 for

Table 5.12: Performance of each F&O strategy and the *irace* configuration on improving the solution obtained by the R&F.

Model	F&O Strategy	Improvement (%)			Time (s)			Infeasibility[A]
		Min	Max	Avg	Min	Max	Avg	Removed
TS	Time Intervals	0.0	13.6	3.4	51.8	3,600.0	915.1	7/9
	Vessel Pairs	0.0	14.6	5.3	3,600.0	3,600.0	3,600.0	2/9
	Interval Vessel	0.0	15.9	4.1	2,069.6	3,600.0	3,516.0	3/9
	Port Types	0.0	4.9	1.1	66.4	910.6	363.6	2/9
	Irace	0.1	15.9	6.4	5,243.4	11,512.3	8,111.9	7/9
FCNF	Time Intervals	0.0	4.6	0.5	0.3	3,600.1	382.0	6/8
	Vessel Pairs	0.0	4.0	0.9	28.8	3,600.0	1,649.5	6/8
	Interval Vessel	0.0	5.7	1.2	550.8	3,600.0	2694.1	7/8
	Port Types	0.0	0.0	0.0	0.1	40.3	6.0	0/8
	Irace	0.0	8.5	1.5	629.4	7,222.1	4,125.7	7/8

Source: From the author (2020).

the TS model and 8 for the FCNF model.

As can be observed in Table 5.12, the improvement of the strategies varied considerably depending on the instance: no improvement occurred for some cases, while for other cases, the improvement was almost 16%. The minor variation was in the port types strategy, which also provided the worse maximum and average improvement. Considering all F&O strategies, the average improvement is higher for the TS model than the FCNF model. Note that higher improvement values of a F&O strategy for some formulation do not necessarily lead to better solutions. It also depends on the quality of the initial solution given by the R&F. Port types strategy was the exception concerning the processing time. It did not reach the given time limit for all tests and provided the worse performance on improving the solutions.

Concerning the number of solutions turned feasible, the TS model's best strategy was the Time Intervals strategy, considering the given time limit. The Vessel Pairs and Interval Vessel strategies consume too much time on average for the TS, obtaining less satisfactory results. For the FCNF, all strategies (except PT) turned feasible most of the infeasible solutions.

5.3.7 Matheuristic Results from the Parameters Calibration

In this section, we present the detailed results after running the *irace* configuration of the F&O. Table 5.13 presents the FCNF model results, while Table 5.14 presents the TS model results, presenting for each instance the processing time and objective function obtained by the R&F algorithm and the F&O strategies. Column *Imp. (%)* presents the

Table 5.13: R&F and F&O results using the FCNF model and the parameter calibration defined by the *irace*.

$\|\mathcal{T}\|$	Instance	R&F		F&O			Total	Gap	BKV	
		Time	Obj	Time	Obj	Improv.	Time		Obj	Time
	LR1_1_DR1_3_VC1_V7a	578.4	-13,272	1,506.9	-13,272	0.0%	2,085.2	0.0%	-13,272[a]	177.0
	LR1_1_DR1_4_VC3_V11a	652.2	-10,858	2,961.5	-11,156	2.7%	3,613.7	0.8%	-11,243[b]	1,578.0
	LR1_1_DR1_4_VC3_V12a	737.8	-10,653	2,130.8	-10,713	0.6%	2,868.7	0.3%	-10,746[d]	7,107.9
	LR1_1_DR1_4_VC3_V12b	1,543.5	-9,059	629.4	-9,060	0.0%	2,172.9	0.3%	-9,085[b]	1,942.0
	LR1_1_DR1_4_VC3_V8a	103.9	-4,936	1,415.9	-5,090	3.1%	1,519.8	0.3%	-5,106[a]	4,609.0
	LR1_1_DR1_4_VC3_V9a	391.3	-6,921	1,059.8	-6,921	0.0%	1,451.1	0.0%	-6,921[c]	1,220.0
	LR1_2_DR1_3_VC2_V6a	542.6	-10,042	1,401.2	-10,096	0.5%	1,943.8	10.3%	-11,134[a]	5,963.0
45	LR1_2_DR1_3_VC3_V8a	152.6	-11,934	864.9	-12,009	0.6%	1,017.5	0.0%	-12,010[a]	7,634.0
	LR2_11_DR2_22_VC3_V6a	1,539.1	27,528	5,353.5	-9,621	-	6,892.7	1.0%	-9,718[a]	8,149.0
	LR2_11_DR2_33_VC4_V11a	3,068.1	-12,870	5,872.8	-13,390	4.0%	8,940.9	4.7%	-14,017[a]	8,913.0
	LR2_11_DR2_33_VC5_V12a	3,134.6	-18,385	4,290.9	-18,524	0.8%	7,425.5	**-0.5%**	-18,423[a]	8,934.0
	LR2_22_DR2_22_VC3_V10a	3,224.8	1,284	6,042.6	-24,985	-	9,267.5	**-0.4%**	-24,874[d]	6,279.5
	LR2_22_DR3_333_VC4_V14a	4,042.3	4,446	7,222.1	-20,066	-	11,264.4	9.4%	-21,952[a]	9,406.0
	LR2_22_DR3_333_VC4_V17a	-	-	-	-	-	-	-	-22,294[c]	10,793.0
	Average	1,516.3		3,134.8		1.2%	4,651.1	2.0%		5,907.5
	LR1_1_DR1_3_VC1_V7a	381.0	-16,675	2,753.8	-16,675	0.0%	3,134.7	0.0%	-16,675[a]	444.0
	LR1_1_DR1_4_VC3_V11a	2,795.8	-12,486	6,710.6	-12,863	3.0%	9,506.4	3.1%	-13,257[a]	7,139.0
	LR1_1_DR1_4_VC3_V12a	1,073.9	-10,991	4,239.3	-11,025	0.3%	5,313.2	0.8%	-11,116[d]	7,731.6
	LR1_1_DR1_4_VC3_V12b	2,583.3	-9,777	4,720.7	-9,796	0.2%	7,304.1	2.6%	-10,053[a]	8,710.0
	LR1_1_DR1_4_VC3_V8a	235.2	10,509	4,852.5	-5,192	-	5,087.7	0.0%	-5,191[a]	7,687.0
	LR1_1_DR1_4_VC3_V9a	234.2	-6,927	1,671.6	-6,927	0.0%	1,905.8	9.0%	-7,552[a]	8,285.0
	LR1_2_DR1_3_VC2_V6a	1,718.9	-12,539	4,743.2	-13,601	8.5%	6,462.1	0.2%	-13,631[a]	8,300.0
60	LR1_2_DR1_3_VC3_V8a	520.9	-14,539	3,985.1	-14,606	0.5%	4,506.1	0.3%	-14,652[a]	8,031.0
	LR2_11_DR2_22_VC3_V6a	3,142.3	4,585	3,992.8	-12,745	-	7,135.1	**-0.7%**	-12,655[a]	8,404.0
	LR2_11_DR2_33_VC4_V11a	5,020.7	3,444	7,210.9	3,364	-	12,231.5	**-**	-15,387[a]	8,943.0
	LR2_11_DR2_33_VC5_V12a	5,027.1	-21,706	7,208.7	-22,107	1.8%	12,235.8	3.8%	-22,948[c]	6,335.0
	LR2_22_DR2_22_VC3_V10a	5,021.8	-9,441	7,208.4	-31,407	-	12,230.2	3.9%	-32,627[a]	8,803.0
	LR2_22_DR3_333_VC4_V14a	-	-	-	-	-	-	-	-26,873[a]	9,542.0
	LR2_22_DR3_333_VC4_V17a	5,234.4	492,125	7,218.2	-24,942	-	12,452.6	8.3%	-27,000[a]	9,615.0
	Average	2,537.7		4,751.1		1.8%	7,107.5	2.6%		7,712.1
	Average All	2,027.0		3,943.0		1.5%	5,879.3	2.3%		6,809.8

Source: From the author (2020).

percentage improvement on the objective function by the F&O when the solution of the R&F is also feasible. Column *Gap* presents the relative gap deviation $\frac{(BKV - Obj)}{Obj} \cdot 100$, where Obj is the final solution obtained by our algorithm and BKV corresponds to the best-known values obtained by Papageorgiou et al. (2014a)[a], Friske and Buriol (2017)[b], Friske and Buriol (2018)[c], and Friske and Buriol (2020)[d]. It is also presented the processing time reported by each cited work in columns *BKV*. When column *Gap* is marked with "-" for some instance, no feasible solution was found without the use of auxiliary variables β_{it}, θ_{it}. If column *Time* is marked with "-" for the R&F algorithm, it could not solve the instance due to abruptly stop in some iteration due to the infeasiblities described in Section 4.3.2.

The final results presented in tables 5.13–5.14 show that the proposed solution approach performs much better for the FCNF model, either in processing time, solution quality, and the number of feasible solutions found. Nevertheless, some good results were also obtained by using the TS model. From the results of Table 5.13, we can observe that:

- The objective function value obtained for the most instances with $|\mathcal{T}| = 45$ was

Table 5.14: R&F and F&O results using the TS model and the parameter calibration defined by the *irace*.

$\|\mathcal{T}\|$	Instance	R&F		F&O			Total	Gap	BKV	
		Time	**Obj**	**Time**	**Obj**	**Improv.**	**Time**		**Obj**	**Time**
45	LR1_1_DR1_3_VC1_V7a	3,235.4	22,728	5,956.9	-13,209	-	9,192.2	0.5%	-13,272[a]	177.0
	LR1_1_DR1_4_VC3_V11a	3,185.2	-10,123	7,925.8	-10,946	8.1%	11,111.0	2.7%	-11,243[b]	1,578.0
	LR1_1_DR1_4_VC3_V12a	3,448.2	-9,938	6,890.9	-10,766	8.3%	10,339.0	**-0.2%**	-10,746[d]	7,107.9
	LR1_1_DR1_4_VC3_V12b	4,213.1	-8,586	8,094.9	-8,653	0.8%	12,308.0	5.0%	-9,085[b]	1,942.0
	LR1_1_DR1_4_VC3_V8a	2,832.7	-4,403	6,946.5	-5,102	15.9%	9,779.2	0.1%	-5,106[a]	4,609.0
	LR1_1_DR1_4_VC3_V9a	3,218.9	-6,402	6,074.3	-6,558	2.4%	9,293.2	5.5%	-6,921[c]	1,220.0
	LR1_2_DR1_3_VC2_V6a	4,204.4	-8,556	7,691.1	-9,901	15.7%	11,895.5	12.5%	-11,134[a]	5,963.0
	LR1_2_DR1_3_VC3_V8a	4,206.9	-10,821	7,479.4	-11,157	3.1%	11,686.3	7.6%	-12,010[a]	7,634.0
	LR2_11_DR2_22_VC3_V6a	-	-	-	-	-	-	-	-9,718[a]	8,149.0
	LR2_11_DR2_33_VC4_V11a	7,039.5	17,552	7,916.3	-12,369	-	14,955.8	13.3%	-14,017[a]	8,913.0
	LR2_11_DR2_33_VC5_V12a	7,140.9	-15,714	8,161.6	-17,287	10.0%	15,302.5	6.6%	-18,423[a]	8,934.0
	LR2_22_DR2_22_VC3_V10a	7,213.1	90,039	9,809.0	-23,350	-	17,022.1	6.5%	-24,874[d]	6,279.5
	LR2_22_DR3_333_VC4_V14a	-	-	-	-	-	-	-	-21,952[a]	9,406.0
	LR2_22_DR3_333_VC4_V17a	8,079.8	26,452	10,428.3	-20,367	-	-	9.5%	-22,294[c]	10,793.0
	Average	4,144.1		6,669.6		8.1%	10,221.9	5.5%		5,907.5
60	LR1_1_DR1_3_VC1_V7a	5,300.6	-16,577	5,243.4	-16,604	0.16%	10,544.0	0.4%	-16,675[a]	444.0
	LR1_1_DR1_4_VC3_V11a	5,135.8	-12,083	7,828.0	-12,101	0.14%	12,963.9	9.6%	-13,257[a]	7,139.0
	LR1_1_DR1_4_VC3_V12a	5,498.7	-10,059	7,521.9	-10,785	7.22%	13,020.6	3.1%	-11,116[d]	7,731.6
	LR1_1_DR1_4_VC3_V12b	6,028.1	-8,440	8,062.7	-9,008	6.73%	14,090.8	11.6%	-10,053[a]	8,710.0
	LR1_1_DR1_4_VC3_V8a	3,472.2	126,398	8,042.9	-4,299	-	11,515.1	20.8%	-5,191[a]	7,687.0
	LR1_1_DR1_4_VC3_V9a	5,343.7	-7,329	7,457.0	-7,338	0.12%	12,800.7	2.9%	-7,552[a]	8,285.0
	LR1_2_DR1_3_VC2_V6a	6,017.8	-11,306	6,524.1	-12,522	10.75%	12,541.8	8.9%	-13,631[a]	8,300.0
	LR1_2_DR1_3_VC3_V8a	6,019.8	20,174	9,278.3	-12,543	-	15,298.1	16.8%	-14,652[a]	8,031.0
	LR2_11_DR2_22_VC3_V6a	-	-	-	-	-	-	-	-12,655[a]	8,404.0
	LR2_11_DR2_33_VC4_V11a	-	-	-	-	-	-	-	-15,387[a]	8,943.0
	LR2_11_DR2_33_VC5_V12a	10,153.3	253,175	10,840.6	34,617	-	20,993.9	-	-22,948[c]	6,335.0
	LR2_22_DR2_22_VC3_V10a	11,024.0	-26,144	10,887.6	-27,071	-	21,911.6	-	-32,627[a]	8,803.0
	LR2_22_DR3_333_VC4_V14a	10,984.9	217,195	11,512.3	-23,522	-	22,497.2	14.2%	-26,873[a]	9,542.0
	LR2_22_DR3_333_VC4_V17a	-	-	-	-	-	-	-	-27,000[a]	9,615.0
	Average	5,355.6		6,657.1		3.6%	12,012.7	9.8%		7,712.1
	Average All	4,749.9		6,663.3		5.8%	11,117.3	7.6%		6,809.8

Source: From the author (2020).

close to the best-known values ($< 1.0\%$).

- For $|\mathcal{T}| = 60$, there are still instances with good results considering the deviation gap, but the tendency is the increase of such value, as well as the higher number of infeasible solutions.

- The processing time for solving smaller instances is relatively higher if compared (disregarding the computer configuration) with the time obtained by other works. It happens mainly due to the lower value defined for the MIP gap as stop criteria in the R&F, extending the processing time at each iteration. When removing the operate-and-depart constraints before starting the F&O, the search space increases, and the F&O iterations tend to increase its consuming time, not necessarily presenting an improvement compared to the case when we keep the operate-and-depart constraints in the F&O.

- Two new best-known values were obtained for $|\mathcal{T}| = 45$ and one for $|\mathcal{T}| = 60$ (gap$< 0.0\%$);

5.4 Metaheuristic Results

This section evaluates our metaheuristic performance and its three components: the multi-start algorithm, the large neighborhood search, and the reduced mixed-integer program model. The objective is to verify our algorithm's capacity to produce good solutions for the MIRP presented, either for instances with short and long planning horizons.

The experiments were carried out in an Intel Core i7 930 computer running at 2.8 GHz with 12 GB RAM. The CPLEX 12.7.1 solver was used to solve the reduced MIP model using a single core. The number of repetitions per run of the algorithm was set to 10, and the average values considering objective function and processing time were considered.

5.4.1 Parameter Calibration

The multi-start algorithm and the LNS parameters must be defined. They were calibrated using the *irace* package[3] for obtaining the best combination of its values. Some parameters have not been calibrated, and a fixed value was set for them. We also study the value variation of the fixed parameters individually. We modified six instances from the test set for defining the training instances for *irace*.The planning horizon was set to $|\mathcal{T}| = 30$, and the x and y coordinates of the ports were changed at random. We selected LR1_1_DR1_3_VC1_V7a, LR1_1_DR1_4_VC3_V11a, LR1_2_DR1_3_VC3_V8a, LR2_22_DR2_22_VC3_V10a, and LR2_22_DR3_333_VC4_V14a as training the instances. In the calibration test, we do not consider the results from the RMIP, i.e., the output is the best solution found after solving the LNS for m solutions. The evaluated parameter values, the best one defined by the calibration, and the parameters with a fixed value are presented in Table 5.15.

The parameter t^{RMIP} is defined according to two characteristics of the instance: the number of loading regions (one or two) and the size of the planning horizon. For the smallest instances corresponding to those with one loading region and $|\mathcal{T}| = 45$, t^{RMIP} is set to 60 seconds. For the instances with two loading regions and the same planning horizon, the value is doubled. For the remaining sizes of $\mathcal{T}$, the time is increased proportionally to the number of time periods.

[3]<http://iridia.ulb.ac.be/irace/>

Table 5.15: Metaheuristic parameter calibration results using the *irace*.

Parameter	Evaluated values	Choosen Value
p^{2nd}	{0%,10%,20%,30%,40%,50%}	0%
p^{CP}	{0%,10%,20%,30%,40%,50%}	50%
f^{start}	[0.9,0.9999]	0.95
f^{end}	[0.0001,0.1]	0.09
c	[0.0001,0.1]	0.01
max^{reset}	{500,1000,2500,5000}	500
n	Fixed	500
m	Fixed	4
$maxIt^{LNS}$	Fixed	25000
$\max^{\text{NA}}$	Fixed	$2(max^{\text{reset}})$
u	Fixed	20%

Source: From the author (2020).

5.4.2 Individual Parameters Analysis

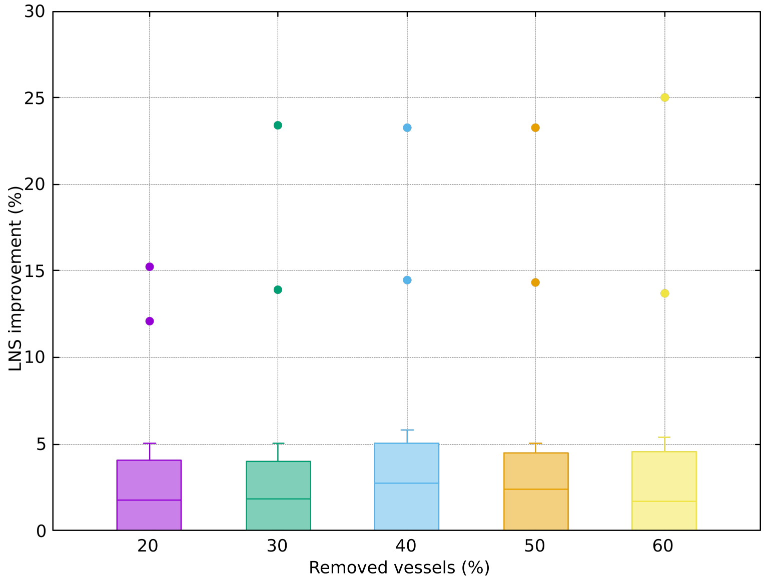
We study how varying separately parameters u, n, and m value affect the proposed method's solution quality and processing time. For these tests, we considered the 14 instances with $|\mathcal{T}| = 45$.

5.4.2.1 Evaluating Parameter u

We first analyze how varying the percentage of vessels removed from the current solution in each LNS iteration affects the solution quality. We performed tests with values $u = \{20, 30, 40, 50, 60\}$. The remaining parameters are fixed to the values defined in the last column of Table 5.15. Figure 5.4 presents the box plot corresponding to the improvement percentage of the multi-start algorithm solutions by the LNS when varying u. Thus, it considers only the instances in which the multi-start algorithm obtained a feasible solution.

As observed in Figure 5.4, the improvement percentage slight varies as parameter u increases considering the median value. The outliers correspond to two instances in which there is a more significant improvement of the solution. Using a binomial statical test with a confidence level of 0.95, we cannot affirm that increasing the value of parameter u provides better LNS solutions. The processing time increases proportionally to the parameter u increase due to the higher number of evaluated vessels in each iteration. Considering the runs in which the multi-start algorithm's solution is infeasible, the LNS was able to remove such infeasibility in 26.3, 28.9, 36.8, 42.1, 47.3 percent of the cases, according to the value of u. Although the feasibility increases with the increase of u (the

Figure 5.4: Improvement of the multi-start algorithm solutions by the LNS varying the values of parameter u.



Source: From the author (2020).

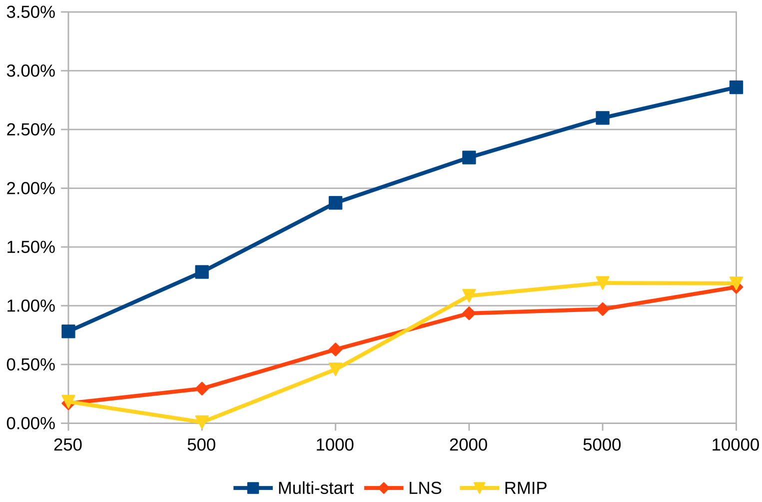
larger parameter u, the larger the search space), it is not possible to confirm such a hypothesis using the same previously mentioned statical test. We decided to use $u = 40$ for the remaining tests, as it provided the best median value considering the improvement of multi-start algorithm solutions.

5.4.2.2 Evaluating Parameter n

The next parameter evaluated was the number of solutions generated by the multi-start algorithm. In this section and the subsequent ones, the results of the RMIP are also considered. We tested the values $n = \{100, 250, 500, 1000, 2000, 5000, 10000\}$ and the results are presented in Figure 5.5, showing for each parameter value the average increase of the solution quality in relation to the base case $n = 100$, considering the multi-start algorithm, the large neighborhood search and the reduced MIP.

The results of Figure 5.5 confirm the tendency that the greater the number of generated solutions, the greater the improvement of the solution quality in the multi-start algorithm. Considering the LNS and the RMIP, we observed several runs of the algorithm that provided worse solutions with $n = 10000$ than with $n = 100$, which contributed to a smaller increase in the solution quality. Although not shown in Figure 5.5, the results demonstrated an approximate increase of 10% in the number of feasible solutions between the minimum and maximum value of n considering the multi-start algorithm and the LNS. For the RMIP, this increase was below 4%, demonstrating that the final algorithm solution

Figure 5.5: Average improvement of the solution quality by varying parameter n.



Source: From the author (2020)

is less affected when varying n value.

We have chosen the value parameter $n = 1000$ for the remainder experiments, as it presented a good balance between solution quality and processing time with a reasonable number of feasible solutions.

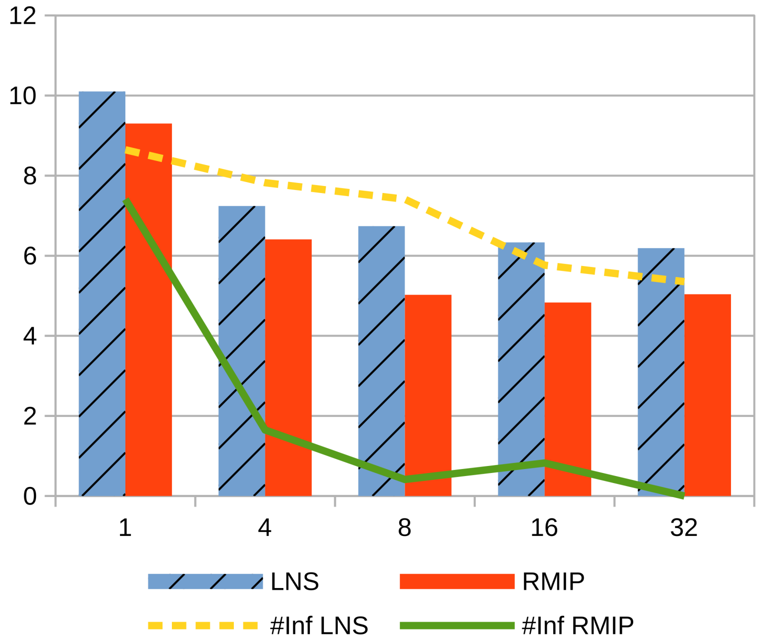
5.4.2.3 Evaluating Parameter m

The last evaluated parameter corresponds to the number of solutions of the multi-start algorithm improved by the LNS and used to build the RMIP. The objective is to verify whether it is valid to increase the m parameter to get a better solution, as it drastically increases the processing time of the LNS. We evaluate the parameter with values $m = \{1, 4, 8, 16, 32\}$. The results are shown in Figure 5.6, presenting for each parameter the average deviation gap from the best-known value considering the feasible solutions found. The lines in the figure represent the average number of instances in which no feasible solution was found.

Figure 5.6 demonstrates a considerably average improvement in the RMIP solutions varying m until value 8, while the average gap of the LNS solutions decreases less than 0.5$\%$ as m is doubled. The time limit for running RMIP may explain the slight difference between the average gap when varying the parameter m from 8 to 32.

The average number of infeasible solutions tends to decrease as m increases, reaching no infeasible solutions with RMIP when $m = 32$. As the processing time increases proportionally to the number m, this parameter should be small. Otherwise, larger instances will take a long time to be solved. Thus, we define $m = 8$ for the remaining

Figure 5.6: Average gap deviation and number of instances with no feasible solution found for different values of parameter m.



Source: From the author (2020).

tests.

5.4.3 Comparison with the Best-known values

We compare our algorithm to the best-known values presented by Papageorgiou et al. (2014a), Friske and Buriol (2017), Friske and Buriol (2018), which provided solutions for $|\mathcal{T}| = \{45, 60\}$. We performed the tests using the parameter values defined in the previous sections. Table 5.16 summarizes the results, presenting for each $|\mathcal{T}|$, the average gap deviation from the best-known value, the average number of instances with no feasible solution found, the average processing time, and the average reduction in the processing time. The detailed results are available in Table A.2 in the appendix. For calculating the processing time reduction, the time was normalized using the *PassMark Software*. The last line of the table presents the average results.

We can observe from Table 5.16 that our approach is not much robust considering solution quality, as the average gap from the best-known value is high, varying from 0.0% to 25.1% depending on the instance. Although not shown in Table 5.16, no feasible solution was found in the 10 repetitions of just one instance with $|\mathcal{T}| = 60$. Our algorithm provided the final solution in a shorter processing time than the other approaches. Thus,

Table 5.16: Comparison between the proposed metaheuristic and the best-known solutions.

$\|\mathcal{T}\|$	*Gap(%)*	*Inf.*	*Time(s)*	*Time reduction (%)*
45	5.1	1.1/14	711.2	79.4%
60	8.0	3.3/14	936.4	91.7%
Avg	6.5		660.7	85.5%

Source: From the author (2020).

we then tested if the results can be improved by increasing the value of some parameters. Thus, we set the parameters u, n, m with the highest values tested in Section 5.4.2 and defined $t^{RMIP} = 3600$ seconds. The results presented an average gap of 3.9% (varying from -0.7% to 13.5%), while there is still one instance in which no feasible solution was found in the ten runs of the algorithm. The algorithm could find new best-known values for five instances considering the individual runs. Although the average processing time was 38.4% faster than the average processing times reported by Papageorgiou et al. (2014a), Friske and Buriol (2017), Friske and Buriol (2018), it is much higher for the small instances, due to the LNS and the RMIP, in which the last usually reaches the time limit of 3600 seconds. Such results are detailed in Table A.3 in the appendix. They demonstrated that the algorithm's performance tends to increase if the parameters u,n,m values are also increased. However, as the processing time required is higher for small instances, it may be more interesting to use higher parameter values to test larger instances and still obtain a lower processing time than other approaches for solving this MIRP.

5.4.4 Long Planning Horizon Instance Results

We ran our metaheuristic in the instances with long planning horizon $|\mathcal{T}| = \{120, 180, 360\}$. For these tests, we perform five repetitions of the algorithm per instance. As this is the first attempt to solve these instances, we evaluate the solution's quality considering the average gap to the lower bound obtained by solving the linear relaxation of the FCNF+ model.

Table 5.17 presents the average results for each size of the planning horizon, showing the number of feasible solutions found and the gap (only for the feasible instances) obtained at the end of the LNS and the end of RMIP. The last column presents the average total time of the algorithm. The detailed results for the feasible solutions obtained are available in Table A.4 in the appendix.

As expected, the difficulty of obtaining feasible solutions and the average gap in-

Table 5.17: LNS and RMIP results for the long planning horizon instances.

	LNS		*RMIP*		*Total time (s)*
	#Feasible	*Gap(%)*	*#Feasible*	*Gap(%)*	
$\|\mathcal{T}\| = 120$	7.0/14	16.4	6.0/14	21,2	1,353.4
$\|\mathcal{T}\| = 180$	5.2/14	19.7	5.2/14	27.0	2,166.6
$\|\mathcal{T}\| = 360$	2.2/14	13.5	2.4/14	13.2	9,512.4

Source: From the author (2020).

creases as the planning horizon's size increases. The multi-start algorithm and the LNS were responsible for 23% and 63% of the processing time on average. The remaining time percentage corresponds to the RMIP processing time. Observe that the processing times increased more than 430% between the tests with $|\mathcal{T}| = 180$ and $|\mathcal{T}| = 360$. This can be explained due to the bottleneck in updating the port's inventory. Every time a vessel operates at a port, the port inventory needs to be updated in each time period from the time of the operation until the end of the planning horizon. As the number of operations performed increases with the planning horizon's size, further inventory updates are required. The RMIP presented worse results considering feasibility and solution quality in general, as the number of variables grows proportionally to the planning horizon size. Thus, the time limit defined for the CPLEX is too restrictive and may be increased to obtain better results with RMIP.

It is important to say that we are not sure if there are feasible solutions for all instances in all long planning horizons tested, in particular, the instances in which spot market variables are necessary to build feasible solutions. This occurs because the value $\alpha_i^{\max}, i \in \mathcal{J}$ representing the total amount of product that can be bought from or sold to a spot market in the planning horizon is the same independently of the size of such planning horizon. Thus, the higher the planning horizon, the more difficult it is to obtain a feasible solution.

To close the chapter, Table 5.18 presents the results obtained by the matheuristic and metaheuristic approaches, presenting the total processing time in seconds and the obtained objective value for the instances with $|\mathcal{T}| = \{45, 60\}$. Columns *TS* and *FCNF+* presents the results obtained by the matheuristic approach using the TS and FCNF+ model with the a priori parameter definition (Sections 5.3.1 and 5.3.2), respectively. Columns *TS-irace* and *FCNF-irace* present the results obtained by the matheuristic approach using the TS and FCNF model with the parameter calibration given by the irace, respectively (Section 5.3.4). Column *Metaheuristic* presents the results of the proposed metaheuristic approach, while column *Metaheuristic+* presents the same approach results, with the highest parameter values tested. For the metaheuristic approaches, the results are the average value of ten runs per instance, considering the results in which a feasible solution was obtained. For each instance, is assigned in **bold** the approach that obtained the best objective value and the lowest processing time.

Table 5.18: Comparison between the matheuristic and metaheuristic approaches proposed

$\|\mathcal{T}\|$	Instance	TS		FCNF+		TS-irace		FCNF-irace		Metaheuristic		Metaheuristic+	
		Time	**Obj**	**Time**	**Obj**	**Time**	**Obj**	**Time**	**Obj**	**Time**	**Obj**	**Time**	**Obj**
45	LR1_1_DR1_3_VC1_V7a	509	**-13,272**	476	**-13,272**	9,192.2	-13,209	2,085.2	**-13,272**	**248.6**	**-13,272**	2,285.6	**-13,272**
	LR1_1_DR1_4_VC3_V11a	981	-11,243	2,792	-11,009	11,111.0	-10,946	3,613.7	-11,156	**838.9**	-11,202	5,497.2	**-11,244**
	LR1_1_DR1_4_VC3_V12a	**662**	-10,372	4,142	-10,709	10,339.0	**-10,766**	2,868.7	-10,713	1,010.7	-10,732	7,107.9	-10,747
	LR1_1_DR1_4_VC3_V12b	1,790	**-9,085**	3,703	-9,028	12,308.0	-8,653	2,172.9	-9,060	**1,159.2**	-9,001	6,650.9	-9,031
	LR1_1_DR1_4_VC3_V8a	**112**	**-5,106**	2,335	-5,060	9,779.2	-5,102	1,519.8	-5,090	523.1	-5,021	2,901.3	-5,104
	LR1_1_DR1_4_VC3_V9a	678	-6,629	1,220	**-6,921**	9,293.2	-6,558	1,451.1	**-6,921**	**551.2**	-6,633	2,278.1	-6,853
	LR1_2_DR1_3_VC2_V6a	2,053	-10,577	2,380	**-10,717**	11,895.5	-9,901	1,943.8	-10,096	**439.3**	-9,748	5,148.0	-9,887
	LR1_2_DR1_3_VC3_V8a	1,443	-11,680	1,202	-11,889	11,686.3	-11,157	1,017.5	**-12,009**	**883.0**	-11,953	9,031.7	-11,984
	LR2_11_DR2_22_VC3_V6a	7,011	-9,550	2,000	-8,510	-	-	6,892.7	**-9,621**	**250.2**	-9,301	1,545.5	-9,498
	LR2_11_DR2_33_VC4_V11a	5,870	**-13,218**	5,340	-11,651	14,955.8	-12,369	8,940.9	-13,390	**642.4**	-12,879	5,524.6	-13,061
	LR2_11_DR2_33_VC5_V12a	5,986	-15,125	4,650	-18,395	15,302.5	-17,287	7,425.5	**-18,524**	**789.3**	-17,788	7,845.6	-18,225
	LR2_22_DR2_22_VC3_V10a	7,471	-21,957	5,890	-24,855	17,022.1	-23,350	9,267.5	-24,985	**991.2**	-24,496	6,279.5	**-24,875**
	LR2_22_DR3_333_VC4_V14a	6,768	25,843	10,285	**-21,925**	-	-	11,264.4	-20,066	**639.6**	-19,525	8,411.6	-20,744
	LR2_22_DR3_333_VC4_V17a	11,439	2,033	10,793	**-22,294**	18,508.1	-20,367	-	-	**990.0**	-18,946	14,060.4	-19,944
60	LR1_1_DR1_3_VC1_V7a	952	**-16,675**	2,922	**-16,675**	10,544.0	-16,604	3,134.7	**-16,675**	**371.2**	**-16,675**	4,588.2	**-16,675**
	LR1_1_DR1_4_VC3_V11a	2,526	-11,516	7,018	-12,155	12,963.9	-12,101	9,506.4	-12,863	**1,298.6**	-12,650	5,676.3	**-13,052**
	LR1_1_DR1_4_VC3_V12a	1,773	**-11,223**	5,768	-11,022	13,020.6	-10,785	5,313.2	-11,025	**1,339.3**	-11,026	7,731.6	-11,116
	LR1_1_DR1_4_VC3_V12b	3,441	-9,958	6,857	-9,830	14,090.8	-9,008	7,304.1	-9,796	**2,145.7**	**-10,016**	7,714.7	-10,006
	LR1_1_DR1_4_VC3_V8a	980	-4,578	4,513	-4,502	11,515.1	-4,299	5,087.7	**-5,192**	**723.7**	-4,150	6,093.7	-4,730
	LR1_1_DR1_4_VC3_V9a	935	-6,904	6,526	-7,328	12,800.7	-7,338	1,905.8	-6,927	**811.1**	-7,092	6,671.5	**-7,404**
	LR1_2_DR1_3_VC2_V6a	2,040	-12,639	4,354	-12,843	12,541.8	-12,522	6,462.1	-13,601	**635.2**	-13,085	7,039.7	**-13,271**
	LR1_2_DR1_3_VC3_V8a	2,068	-14,329	4,866	-14,152	15,298.1	-12,543	4,506.1	**-14,606**	**1,428.9**	-14,424	8,937.9	-14,526
	LR2_11_DR2_22_VC3_V6a	5,648	39,102	4,793	42,126	-	-	7,135.1	**-12,745**	**512.4**	69,589	7,005.5	25,128
	LR2_11_DR2_33_VC4_V11a	6,262	181,868	8,595	-14,379	-	-	12,231.5	3,364	**476.5**	-7,883	7,196.1	**-14,608**
	LR2_11_DR2_33_VC5_V12a	6,304	176,226	6,335	**-22,948**	20,993.9	34,617	12,235.8	-22,107	**648.1**	5,272	10,055.5	-22,039
	LR2_22_DR2_22_VC3_V10a	6,462	826,372	9,186	**-31,598**	21,911.6	-27,071	12,230.2	-31,407	**659.7**	-26,048	8,098.5	**-30,833**
	LR2_22_DR3_333_VC4_V14a	7,316	2,218,890	12,716	**-25,069**	22,497.2	-23,522	-	-	**829.0**	2,900	10,754.8	-16,430
	LR2_22_DR3_333_VC4_V17a	7,866	1,110,680	15,020	**-25,236**	-	-	12,452.6	-24,942	**1,230.7**	-23,196	16,176.0	-23,717

Source: From the author (2020).

From the Table 5.18 we can conclude the following:

- The approaches that obtained the major number of best solutions considering the objective function value were the *FCNF+*, the *FCNF* and the *Metaheuristic+*. Note that the other approaches still were able to obtain the best solution for some cases.
- The *Metaheuristic* is able to obtain solutions in shorter processing time in most cases. However, if it is desired to obtain a better solution increasing the algorithm parameter values, the processing time increases considerably.
- For the smaller instances (one loading region - LR1), it is preferable to use the simplest version of the matheuristic (results of *TS*) and the metaheuristic (results of *Metaheuristic*), as they provided good solutions in shorter processing times.
- For larger instances (two loading regions - LR2), it is preferable to use the matheuristic approach with the FCNF+ model or the metaheuristic with the highest parameter values, as they provided the major number of feasible solutions, besides the best results considering objective value.

6 CONCLUDING REMARKS

This work proposed and explored different solutions approaches for a Maritime Inventory Routing Problem variant proposed in the literature. The MIRP is a challenging problem that has gained attention in the last decades and consists of routing and scheduling a fleet of heterogeneous vessels for transporting products between production and consumption ports, where inventory management is taken into consideration. Two main solutions approaches were presented: a matheuristic framework that combines exact and heuristic methods and a metaheuristic approach independent of a mathematical solver, which to the best of our knowledge, was the first one proposed for the MIRP variant studied here.

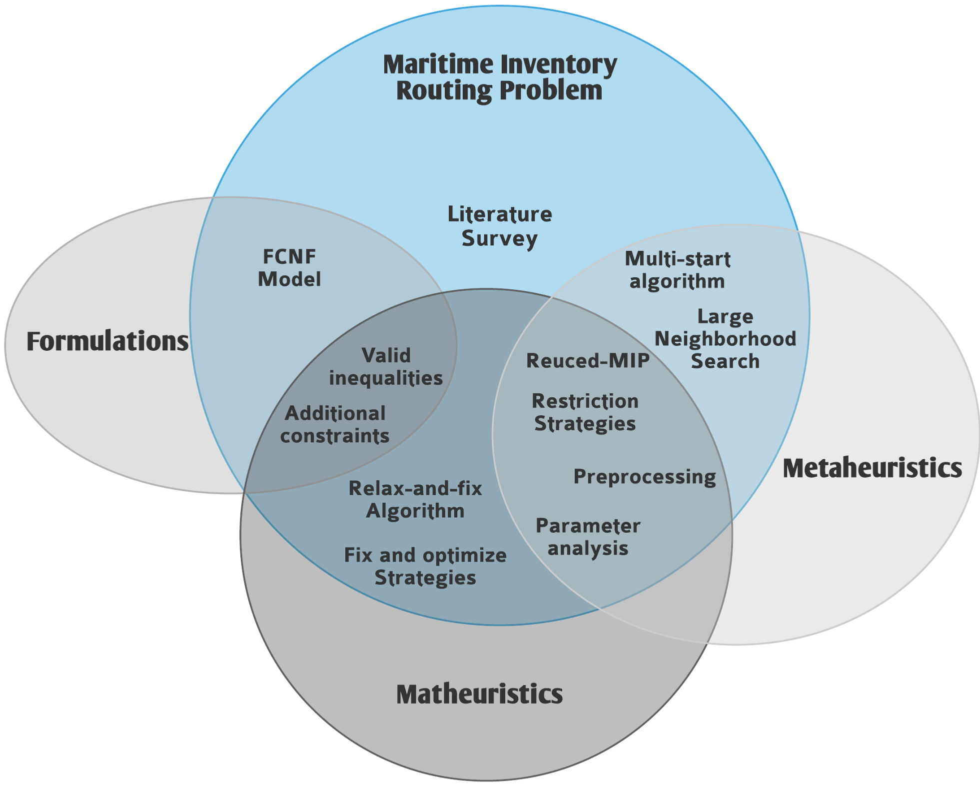
The matheuristic framework comprises a relax and fix algorithm for building an initial solution and a fix and optimize algorithm as the improvement phase. They were evaluated over two discrete-time formulations: a standard time-space network and a fixed-charge network flow. Differents formulation components were proposed and evaluated to demonstrate their contribution to the performance of the algorithms.

The metaheuristic method is composed of a multi-start algorithm to build initial solutions and a large neighborhood search to improve them. Also, a reduced mixed-integer program was proposed to work as a post-optimization strategy. The method could provide solutions for large planning horizon instances in which no results were presented in the literature.

Computational experiments were performed on public instances to analyze both methods' solution quality and processing times varying the parameter values. Besides test with a priori definition of parameter values for the algorithms, we also used an automatic configuration tool to define a more robust configuration to obtain better quality solutions on average. The results demonstrated that both methods are useful for solving the problem instances, as they could obtain good solutions in relatively short processing times, besides new best-known values. The metaheuristic was able to obtain feasible solutions for large problem instances, which considers long planning horizons. Nevertheless, obtaining even feasible solutions for this problem is challenging due to the port inventory constraints.

Figure 6.1 summarizes through a Venn diagram the contribution of this thesis considering four key sets: MIRP, formulations, matheuristic, and metaheuristic. Each word in the sets represents an entirely new contribution or an adapted for this work based on other works. Also, observe that the intersection between the sets represents contributions

Figure 6.1: Contributions of this thesis.



Source: From the author (2020).

that are related to the sets. For example, the *Restrictions Strategies* are applicable for both matheuristics and metaheuristics.

6.1 Future Work

Several directions can be pointed out as future work considering the MIRP and the solutions methods. For the problem formulation, a possible direction is to convert the discrete-time model to a continuous-time model, which usually has a smaller number of discrete variables for the MIRP. Thus, exact methods or matheuristics as the R&F and F&O can be more effective in solving the problem. Nevertheless, some issues need to be handled, such as the variable production and consumption rates at the loading and discharging ports, presented to the best of our knowledge in discrete-time formulations only.

Another possible study is to evaluate how the instance characteristics affect solving the problem, either directly by a mathematical solver or by the proposed methods. For example, besides the problem allows multiple berths (parameter B_i), all MIRPLIB instances used in the computational experiments have $B_i = 1$, for all $i \in \mathcal{J}$. Thus, if such parameter value is increased (as expected, some ports have several berths), we can evaluate how they affect the methods considering aspects as solution quality and feasibility.

Throughout this work, we observed that the algorithms' performance depends on the parameters' suitable choice, which depends on the instance. A possible direction to handle the parameter issues is instead of using *irace* to define one algorithm configuration for all the instances, is to use an algorithm selection technique, such as the Autofolio (LINDAUER et al., 2015), to define the best parameter configuration for each problem instance. In such a technique, besides the algorithm parameters and the execution results, instance features are evaluated by a machine learning model. After a training phase, the model can define the best algorithm parameter configuration for solving not seen instances according to their features. However, for this approach to be effective, an extensive training set should be necessary, not the case for the MIRPs in general, as the instances based on real data usually are not publicly available.

A strategy that can be considered for the matheuristic approach is to use the feasibility pump heuristic (FISCHETTI; GLOVER; LODI, 2005) at the R&F iterations to build good initial solutions and possibly reducing the number of infeasible solutions. Such a strategy was used in Agra et al. (2014) to solve the rolling horizon heuristic iterations. It is also desirable to avoid the use of auxiliary variables β_{it} and θ_{it} in the solution. When some of the constructive algorithms need to use such variables, it is not guaranteed that the improvement strategies will turn the solution feasible, i.e., setting all auxiliary variables to zero. An alternative approach that can be considered in both matheuristic and metaheuristic approaches is after building a feasible solution that has some auxiliary variable with a positive value, change the problem objective to exclusively minimize the number of β_{it} and θ_{it} variables. This problem can then be solved with a mathematical solver for a limited time before starting the improvement phase.

For the metaheuristic approach, some improvements can be suggested: the first is to improve the multi-start algorithm using memory mechanisms to guide the search based on previously built solutions. Another improvement on the method is to use a small MIP model to define the time and the best quantity of product to be operated in each port visit by a vessel (Section 4.5.7). We can also evolve the LNS algorithm to an adaptive large neighborhood search in which different neighborhoods can be explored, besides defining more robust selection criteria based on specific rules instead of random selection. Finally, combinations between the matheuristic and metaheuristic methods can be tested. We can consider using the multi-start algorithm to generate several solutions to warm starts each R&F iteration. Such an approach can improve algorithm efficiency, leading to feasible solutions faster. It may also be interesting to use more than one solution generated at R&F iterations to provide several solutions at the end of the algorithm. Such an approach can be parallelized using multiple processing cores. Another combination that can be considered is to use the F&O strategies to improve a set of solutions generated by the multi-start algorithm.

REFERENCES

AGRA, A.; ANDERSSON, H.; CHRISTIANSEN, M.; WOLSEY, L. A maritime inventory routing problem: Discrete time formulations and valid inequalities. **Networks**, v. 62, n. 4, p. 297–314, 2013.

AGRA, A.; CHRISTIANSEN, M.; DELGADO, A. Mixed Integer Formulations for a Short Sea Fuel Oil Distribution Problem. **Transportation Science**, INFORMS, v. 47, n. 1, p. 108–124, 2013.

AGRA, A.; CHRISTIANSEN, M.; DELGADO, A. Discrete time and continuous time formulations for a short sea inventory routing problem. **Optimization and Engineering**, Springer, v. 18, n. 1, p. 269–297, 2017.

AGRA, A.; CHRISTIANSEN, M.; DELGADO, A.; SIMONETTI, L. Hybrid heuristics for a short sea inventory routing problem. **European Journal of Operational Research**, v. 236, n. 3, p. 924 – 935, 2014, vehicle Routing and Distribution Logistics.

AGRA, A.; CHRISTIANSEN, M.; DELGADO, A.; HVATTUM, L. M. A maritime inventory routing problem with stochastic sailing and port times. **Computers & Operations Research**, v. 61, p. 18 – 30, 2015.

AGRA, A.; CHRISTIANSEN, M.; HVATTUM, L. M.; RODRIGUES, F. A mip based local search heuristic for a stochastic maritime inventory routing problem. In: SPRINGER. **International Conference on Computational Logistics**. [S.l.], 2016. p. 18–34.

AGRA, A.; CHRISTIANSEN, M.; HVATTUM, L. M.; RODRIGUES, F. Robust optimization for a maritime inventory routing problem. **Transportation Science**, INFORMS, v. 52, n. 3, p. 509–525, 2018.

AGRA, A.; CHRISTIANSEN, M.; IVARSØY, K. S.; SOLHAUG, I. E.; TOMASGARD, A. Combined ship routing and inventory management in the salmon farming industry. **Annals of Operations Research**, Springer, v. 253, n. 2, p. 799–823, 2017.

AIZEMBERG, L.; KRAMER, H. H.; PESSOA, A. A.; UCHOA, E. Formulations for a problem of petroleum transportation. **European Journal of Operational Research**, Elsevier, v. 237, n. 1, p. 82–90, 2014.

AKARTUNALI, K.; MILLER, A. J. A heuristic approach for big bucket multi-level production planning problems. **European Journal of Operational Research**, Elsevier, v. 193, n. 2, p. 396–411, 2009.

AL-KHAYYAL, F.; HWANG, S.-J. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part i: Applications and model. **European Journal of Operational Research**, Elsevier, v. 176, n. 1, p. 106–130, 2007.

ANDERSSON, H. A maritime pulp distribution problem. **INFOR: Information Systems and Operational Research**, Taylor & Francis, v. 49, n. 2, p. 125–138, 2011.

ANDERSSON, H.; CHRISTIANSEN, M.; DESAULNIERS, G. A new decomposition algorithm for a liquefied natural gas inventory routing problem. **International Journal of Production Research**, Taylor & Francis, v. 54, n. 2, p. 564–578, 2016.

ANDERSSON, H.; HOFF, A.; CHRISTIANSEN, M.; HASLE, G.; LØKKETANGEN, A. Industrial aspects and literature survey: Combined inventory management and routing. **Computers & Operations Research**, Elsevier, v. 37, n. 9, p. 1515–1536, 2010.

ANTAQ, A. N. d. T. A. **Anuário 2017**. 2017. <http://portal.antaq.gov.br/wp-content/uploads/2018/02/20180112_Anu%C3%A1rio_2017_v4-4-vers%C3%A3o-final.pdf>. Accessed: 2018-06-10.

APPELGREN, L. H. A column generation algorithm for a ship scheduling problem. **Transportation Science**, INFORMS, v. 3, n. 1, p. 53–68, 1969.

APPELGREN, L. H. Integer programming methods for a vessel scheduling problem. **Transportation Science**, INFORMS, v. 5, n. 1, p. 64–78, 1971.

ARAUJO, S. A. de; ARENALES, M. N.; CLARK, A. R. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. **Journal of Heuristics**, Springer, v. 13, n. 4, p. 337–358, 2007.

BELLMORE, M.; BENNINGTON, G.; LUBORE, S. A multivehicle tanker scheduling problem. **Transportation Science**, INFORMS, v. 5, n. 1, p. 36–47, 1971.

BERALDI, P.; GHIANI, G.; GRIECO, A.; GUERRIERO, E. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. **Computers & Operations Research**, Elsevier, v. 35, n. 11, p. 3644–3656, 2008.

BILGEN, B.; OZKARAHAN, I. A mixed-integer linear programming model for bulk grain blending and shipping. **International journal of production economics**, Elsevier, v. 107, n. 2, p. 555–571, 2007.

BRISKIN, L. E. Selecting delivery dates in the tanker scheduling problem. **Management Science**, INFORMS, v. 12, n. 6, p. B224–B235, 1966.

CHRISTIANSEN, M. Decomposition of a combined inventory and time constrained ship routing problem. **Transportation science**, INFORMS, v. 33, n. 1, p. 3–16, 1999.

CHRISTIANSEN, M.; FAGERHOLT, K. Chapter 13: Ship routing and scheduling in industrial and tramp shipping. In: **Vehicle Routing: Problems, Methods, and Applications, Second Edition**. [S.l.]: SIAM, 2014. p. 381–408.

CHRISTIANSEN, M.; FAGERHOLT, K.; FLATBERG, T.; HAUGEN, Ø.; KLOSTER, O.; LUND, E. H. Maritime inventory routing with multiple products: A case study from the cement industry. **European Journal of Operational Research**, Elsevier, v. 208, n. 1, p. 86–94, 2011.

CHRISTIANSEN, M.; FAGERHOLT, K.; NYGREEN, B.; RONEN, D. Maritime transportation. **Handbooks in operations research and management science**, Elsevier, v. 14, p. 189–284, 2007.

CHRISTIANSEN, M.; FAGERHOLT, K.; NYGREEN, B.; RONEN, D. Ship routing and scheduling in the new millennium. **European Journal of Operational Research**, Elsevier, v. 228, n. 3, p. 467–483, 2013.

CHRISTIANSEN, M.; FAGERHOLT, K.; RONEN, D. Ship routing and scheduling: Status and perspectives. **Transportation science**, INFORMS, v. 38, n. 1, p. 1–18, 2004.

CHRISTIANSEN, M.; NYGREEN, B. A method for solving ship routing problems with inventory constraints. **Annals of operations research**, Springer, v. 81, p. 357–378, 1998.

CHRISTIANSEN, M.; NYGREEN, B. Modelling path flows for a combined ship routingand inventory management problem. **Annals of Operations Research**, Springer, v. 82, p. 391–413, 1998.

COELHO, L. C.; CORDEAU, J.-F.; LAPORTE, G. Thirty years of inventory routing. **Transportation Science**, INFORMS, v. 48, n. 1, p. 1–19, 2013.

DANTZIG, G. B.; FULKERSON, D. R. Minimizing the number of tankers to meet a fixed schedule. **Naval Research Logistics Quarterly**, Wiley Online Library, v. 1, n. 3, p. 217–222, 1954.

DAUZÈRE-PÉRÈS, S.; NORDLI, A.; OLSTAD, A.; HAUGEN, K.; KOESTER, U.; OLAV, M. P.; TEISTKLUB, G.; REISTAD, A. Omya hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to european paper manufacturers. **Interfaces**, Informs, v. 37, n. 1, p. 39–51, 2007.

DE, A.; KUMAR, S. K.; GUNASEKARAN, A.; TIWARI, M. K. Sustainable maritime inventory routing problem with time window constraints. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 61, p. 77–95, 2017.

DILLENBERGER, C.; ESCUDERO, L. F.; WOLLENSAK, A.; ZHANG, W. On practical resource allocation for production planning and scheduling with period overlapping setups. **European Journal of Operational Research**, Elsevier, v. 75, n. 2, p. 275–286, 1994.

DIZ, G. S. d. S.; OLIVEIRA, F.; HAMACHER, S. Improving maritime inventory routing: application to a brazilian petroleum case. **Maritime Policy & Management**, Taylor & Francis, v. 44, n. 1, p. 42–61, 2017.

ENGINEER, F. G.; FURMAN, K. C.; NEMHAUSER, G. L.; SAVELSBERGH, M. W.; SONG, J.-H. A branch-price-and-cut algorithm for single-product maritime inventory routing. **Operations Research**, INFORMS, v. 60, n. 1, p. 106–122, 2012.

FEDERGRUEN, A.; MEISSNER, J.; TZUR, M. Progressive interval heuristics for multi-item capacitated lot-sizing problems. **Operations Research**, INFORMS, v. 55, n. 3, p. 490–502, 2007.

FERREIRA, D.; MORABITO, R.; RANGEL, S. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. **European Journal of Operational Research**, Elsevier, v. 196, n. 2, p. 697–706, 2009.

FISCHETTI, M.; GLOVER, F.; LODI, A. The feasibility pump. **Mathematical Programming**, Springer, v. 104, n. 1, p. 91–104, 2005.

FLOOD, M. M. Application of transportation theory to scheduling a military tanker fleet. **Journal of the Operations Research Society of America**, INFORMS, v. 2, n. 2, p. 150–162, 1954.

FODSTAD, M.; UGGEN, K. T.; RØMO, F.; LIUM, A.-G.; STREMERSCH, G.; HECQ, S. LNGScheduler: a rich model for coordinating vessel routing, inventories and trade in the liquefied natural gas supply chain. **The journal of energy markets**, Incisive Media Limited, v. 3, n. 4, p. 31–64, 2010.

FOSS, E.; MYKLEBUST, T. N.; ANDERSSON, H.; CHRISTIANSEN, M. A multi-product maritime inventory routing problem with undedicated compartments. In: SPRINGER. **International Conference on Computational Logistics**. [S.l.], 2016. p. 3–17.

FRISKE, M. W.; BURIOL, L. S. A relax-and-fix algorithm for a maritime inventory routing problem. In: SPRINGER. **International Conference on Computational Logistics**. [S.l.], 2017. p. 270–284.

FRISKE, M. W.; BURIOL, L. S. Applying a relax-and-fix approach to a fixed charge network flow model of a maritime inventory routing problem. In: SPRINGER. **International Conference on Computational Logistics**. [S.l.], 2018. p. 3–16.

FRISKE, M. W.; BURIOL, L. S. A multi-start algorithm and a large neighborhood search for a maritime inventory routing problem. In: IEEE. **2020 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2020. p. 1–8.

FRISKE, M. W.; BURIOL, L. S.; CAMPONOGARA, E. A relax-and-fix and fix-and-optimize algorithm for a maritime inventory routing problem. Submited for review to Computers & Operations Research. 2020.

FURMAN, K. C.; SONG, J.-H.; KOCIS, G. R.; MCDONALD, M. K.; WARRICK, P. H. Feedstock routing in the exxonmobil downstream sector. **Interfaces**, INFORMS, v. 41, n. 2, p. 149–163, 2011.

GOEL, V.; FURMAN, K. C.; SONG, J.-H.; EL-BAKRY, A. S. Large neighborhood search for lng inventory routing. **Journal of Heuristics**, Springer, v. 18, n. 6, p. 821–848, 2012.

GOEL, V.; SLUSKY, M.; HOEVE, W.-J. van; FURMAN, K. C.; SHAO, Y. Constraint programming for lng ship scheduling and inventory management. **European Journal of Operational Research**, Elsevier, v. 241, n. 3, p. 662–673, 2015.

GRØNHAUG, R.; CHRISTIANSEN, M. Supply chain optimization for the liquefied natural gas business. In: **Innovations in distribution logistics**. [S.l.]: Springer, 2009. p. 195–218.

GRØNHAUG, R.; CHRISTIANSEN, M.; DESAULNIERS, G.; DESROSIERS, J. A branch-and-price method for a liquefied natural gas inventory routing problem. **Transportation Science**, INFORMS, v. 44, n. 3, p. 400–415, 2010.

HEMMATI, A.; HVATTUM, L. M.; CHRISTIANSEN, M.; LAPORTE, G. An iterative two-phase hybrid matheuristic for a multi-product short sea inventory-routing problem. **European Journal of Operational Research**, Elsevier, v. 252, n. 3, p. 775–788, 2016.

HEMMATI, A.; STÅLHANE, M.; HVATTUM, L. M.; ANDERSSON, H. An effective heuristic for solving a combined cargo and inventory routing problem in tramp shipping. **Computers & Operations Research**, Elsevier, v. 64, p. 274–282, 2015.

HEWITT, M.; NEMHAUSER, G.; SAVELSBERGH, M.; SONG, J.-H. A branch-and-price guided search approach to maritime inventory routing. **Computers & Operations Research**, Elsevier, v. 40, n. 5, p. 1410–1419, 2013.

JIANG, Y.; GROSSMANN, I. E. Alternative mixed-integer linear programming models of a maritime inventory routing problem. **Computers & Chemical Engineering**, Elsevier, v. 77, p. 147–161, 2015.

LANG, J. C.; SHEN, Z.-J. M. Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. **European Journal of Operational Research**, Elsevier, v. 214, n. 3, p. 595–605, 2011.

LAWRENCE, S. A. **International sea transport: the years ahead**. [S.l.]: Lexington Books, 1972.

LI, J.; KARIMI, I. A.; SRINIVASAN, R. Efficient bulk maritime logistics for the supply and delivery of multiple chemicals. **Computers & Chemical Engineering**, Elsevier, v. 34, n. 12, p. 2118–2128, 2010.

LINDAUER, M.; HOOS, H. H.; HUTTER, F.; SCHAUB, T. Autofolio: An automatically configured algorithm selector. **Journal of Artificial Intelligence Research**, v. 53, p. 745–778, 2015.

LÓPEZ-IBÁÑEZ, M.; DUBOIS-LACOSTE, J.; CÁCERES, L. P.; BIRATTARI, M.; STÜTZLE, T. The irace package: Iterated racing for automatic algorithm configuration. **Operations Research Perspectives**, Elsevier, v. 3, p. 43–58, 2016.

MARTÍ, R.; MORENO-VEGA, J. M.; DUARTE, A. Advanced multi-start methods. In: **Handbook of metaheuristics. International Series in Operations Research & Management Science**. [S.l.]: Springer, 2010. v. 146, p. 265–281.

MCKAY, M.; HARTLEY, H. Computerized scheduling of seagoing tankers. **Naval Research Logistics (NRL)**, Wiley Online Library, v. 21, n. 2, p. 255–264, 1974.

MILLER, D. M. An interactive, computer-aided ship scheduling system. **European Journal of Operational Research**, Elsevier, v. 32, n. 3, p. 363–379, 1987.

MOHAMMADI, M.; GHOMI, S. F.; KARIMI, B.; TORABI, S. A. Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. **Journal of Intelligent Manufacturing**, Springer, v. 21, n. 4, p. 501–510, 2010.

MUNGUÍA, L.-M.; AHMED, S.; BADER, D. A.; NEMHAUSER, G. L.; SHAO, Y.; PAPAGEORGIOU, D. J. Tailoring parallel alternating criteria search for domain specific MIP: Application to maritime inventory routing. **Computers & Operations Research**, Elsevier, v. 111, p. 21–34, 2019.

MUTLU, F.; MSAKNI, M. K.; YILDIZ, H.; SÖNMEZ, E.; POKHAREL, S. A comprehensive annual delivery program for upstream liquefied natural gas supply chain. **European Journal of Operational Research**, Elsevier, v. 250, n. 1, p. 120–130, 2016.

PAPAGEORGIOU, D. J. **Maritime Inventory Routing Problem Library (MIRPLIB)**. 2013. <https://mirplib.scl.gatech.edu/>. Accessed: 2020-11-30.

PAPAGEORGIOU, D. J.; CHEON, M.-S.; HARWOOD, S.; TRESPALACIOS, F.; NEMHAUSER, G. L. Recent progress using matheuristics for strategic maritime inventory routing. In: **Modeling, Computing and Data Handling Methodologies for Maritime Transportation**. [S.l.]: Springer, 2018. p. 59–94.

PAPAGEORGIOU, D. J.; CHEON, M.-S.; NEMHAUSER, G.; SOKOL, J. Approximate dynamic programming for a class of long-horizon maritime inventory routing problems. **Transportation Science**, INFORMS, v. 49, n. 4, p. 870–885, 2015.

PAPAGEORGIOU, D. J.; KEHA, A. B.; NEMHAUSER, G. L.; SOKOL, J. Two-stage decomposition algorithms for single product maritime inventory routing. **INFORMS Journal on Computing**, INFORMS, v. 26, n. 4, p. 825–847, 2014.

PAPAGEORGIOU, D. J.; NEMHAUSER, G. L.; SOKOL, J.; CHEON, M.-S.; KEHA, A. B. Mirplib–a library of maritime inventory routing problem instances: Survey, core model, and benchmark results. **European Journal of Operational Research**, Elsevier, v. 235, n. 2, p. 350–366, 2014.

PERSSON, J. A.; GÖTHE-LUNDGREN, M. Shipment planning at oil refineries using column generation and valid inequalities. **European Journal of Operational Research**, Elsevier, v. 163, n. 3, p. 631–652, 2005.

PISINGER, D.; ROPKE, S. Large neighborhood search. In: **Handbook of metaheuristics**. [S.l.]: Springer, 2010. p. 399–419.

POCHET, Y.; WARICHET, F. A tighter continuous time formulation for the cyclic scheduling of a mixed plant. **Computers & Chemical Engineering**, Elsevier, v. 32, n. 11, p. 2723–2744, 2008.

POCHET, Y.; WOLSEY, L. A. **Production planning by mixed integer programming**. [S.l.]: Springer Science & Business Media, 2006.

PSARAFTIS, H. N. Ship routing and scheduling: the cart before the horse conjecture. **Maritime Economics & Logistics**, Springer, v. 21, n. 1, p. 111–124, 2019.

PSARAFTIS, H. N.; KONTOVAS, C. A. Speed models for energy-efficient maritime transportation: A taxonomy and survey. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 26, p. 331–351, 2013.

RAKKE, J. G.; STÅLHANE, M.; MOE, C. R.; CHRISTIANSEN, M.; ANDERSSON, H.; FAGERHOLT, K.; NORSTAD, I. A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 19, n. 5, p. 896–911, 2011.

ROCHA, R.; GROSSMANN, I. E.; ARAGÃO, M. V. P. de. Cascading knapsack inequalities: reformulation of a crude oil distribution problem. **Annals of Operations Research**, Springer, v. 203, n. 1, p. 217–234, 2013.

RODRIGUES, F.; AGRA, A.; CHRISTIANSEN, M.; HVATTUM, L. M.; REQUEJO, C. Comparing techniques for modelling uncertainty in a maritime inventory routing problem. **European Journal of Operational Research**, Elsevier, v. 277, n. 3, p. 831–845, 2019.

RONEN, D. **Scheduling of vessels for shipment of bulk and semi-bulk commodities originating in a single area**. Thesis (PhD) — The Ohio State University, 1979.

RONEN, D. Cargo ships routing and scheduling: Survey of models and problems. **European Journal of Operational Research**, Elsevier, v. 12, n. 2, p. 119–126, 1983.

RONEN, D. Ship scheduling: The last decade. **European Journal of Operational Research**, Elsevier, v. 71, n. 3, p. 325–333, 1993.

RONEN, D. Marine inventory routing: Shipments planning. **Journal of the Operational Research Society**, Taylor & Francis, v. 53, n. 1, p. 108–114, 2002.

ROPKE, S.; PISINGER, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. **Transportation science**, Informs, v. 40, n. 4, p. 455–472, 2006.

SHEN, Q.; CHU, F.; CHEN, H. A lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. **Computers & Chemical Engineering**, Elsevier, v. 35, n. 10, p. 2113–2123, 2011.

SISWANTO, N.; ESSAM, D.; SARKER, R. Solving the ship inventory routing and scheduling problem with undedicated compartments. **Computers & Industrial Engineering**, Elsevier, v. 61, n. 2, p. 289–299, 2011.

SONG, J.-H.; FURMAN, K. C. A maritime inventory routing problem: Practical approach. **Computers & Operations Research**, Elsevier, v. 40, n. 3, p. 657–665, 2013.

STADTLER, H. Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. **Operations Research**, INFORMS, v. 51, n. 3, p. 487–502, 2003.

STANZANI, A. d. L.; PUREZA, V.; MORABITO, R.; SILVA, B. J. V. d.; YAMASHITA, D.; RIBAS, P. C. Optimizing multiship routing and scheduling with constraints on inventory levels in a brazilian oil company. **International Transactions in Operational Research**, Wiley Online Library, v. 25, n. 4, p. 1163–1198, 2018.

STOTT, K. L.; DOUGLAS, B. W. A model-based decision support system for planning and scheduling ocean-borne transportation. **Interfaces**, INFORMS, v. 11, n. 4, p. 1–10, 1981.

TAILLARD, É. D.; VOSS, S. POPMUSIC—partial optimization metaheuristic under special intensification conditions. In: **Essays and surveys in metaheuristics**. [S.l.]: Springer, 2002. p. 613–629.

TOLEDO, C. F. M.; ARANTES, M. da S.; HOSSOMI, M. Y. B.; FRANÇA, P. M.; AKARTUNALI, K. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. **Journal of heuristics**, Springer, v. 21, n. 5, p. 687–717, 2015.

UGGEN, K. T.; FODSTAD, M.; NØRSTEBØ, V. S. Using and extending fix-and-relax to solve maritime inventory routing problems. **Top**, Springer, v. 21, n. 2, p. 355–377, 2013.

UNCTAD. **Review of Maritime Transport 2019**. [S.l.]: United Nations Publications, 2019.

WOLSEY, L. A. **Integer programming**. [S.l.]: John Wiley & Sons, 1998.

ZHANG, C.; NEMHAUSER, G.; SOKOL, J.; PAPAGEORGIOU, M.-s. C. D. Robust Inventory Routing with Flexible Time Window Allocation. 2015.

APPENDIX A — COMPUTATIONAL RESULTS INDIVIDUALIZED BY INSTANCE

Table A.1: Lower bound values considering different formulations.

$\|\mathcal{T}\|$	Instance	TS	TS+VI	Optimistic System Model	FCNF	FCNF + VI	SystemModel-2Port
45	LR1_1_DR1_3_VC1_V7a	-25,646.6	-18,068.3	-14,410.0	-14,291.6	-14,205.6	-13,273.0
	LR1_1_DR1_4_VC3_V11a	-29,741.0	-21,766.9	-12,994.0	-11,821.3	-11,689.4	-11,289.0
	LR1_1_DR1_4_VC3_V12a	-33,397.4	-22,998.2	-12,329.0	-11,361.4	-11,315.3	-10,739.0
	LR1_1_DR1_4_VC3_V12b	-30,645.7	-21,412.8	-9,578.0	-9,752.2	-9,688.7	-9,073.0
	LR1_1_DR1_4_VC3_V8a	-24,312.8	-15,413.2	-6,153.0	-5,524.1	-5,468.9	-5,174.0
	LR1_1_DR1_4_VC3_V9a	-24,759.0	-16,953.4	-8,242.0	-7,469.6	-7,402.0	-6,959.0
	LR1_2_DR1_3_VC2_V6a	-23,225.0	-21,734.9	-12,763.0	-11,741.5	-11,729.7	-11,146.0
	LR1_2_DR1_3_VC3_V8a	-28,164.0	-26,088.2	-13,625.0	-12,184.9	-12,137.7	-12,012.0
	LR2_11_DR2_22_VC3_V6a	-23,744.2	-17,332.1	-10,802.0	-12,423.1	-12,417.3	-9,779.0
	LR2_11_DR2_33_VC4_V11a	-36,104.8	-26,617.9	-16,445.0	-15,793.2	-15,761.3	-15,137.0
	LR2_11_DR2_33_VC5_V12a	-42,973.1	-31,823.3	-20,668.0	-19,714.7	-19,634.7	-19,092.0
	LR2_22_DR2_22_VC3_V10a	-45,118.7	-42,000.2	-27,803.0	-28,032.6	-27,826.6	-25,576.0
	LR2_22_DR3_333_VC4_V14a	-49,052.2	-46,646.5	-27,216.0	-25,764.8	-25,637.1	-23,967.0
	LR2_22_DR3_333_VC4_V17a	-55,568.5	-51,574.4	-27,628.0	-25,856.2	-25,853.0	-23,553.0
60	LR1_1_DR1_3_VC1_V7a	-33,347.6	-21,535.7	-17,847.0	-17,662.7	-17,556.4	-16,676.0
	LR1_1_DR1_4_VC3_V11a	-38,827.6	-25,103.8	-15,020.0	-13,749.1	-13,642.1	-13,383.0
	LR1_1_DR1_4_VC3_V12a	-43,537.5	-26,423.4	-12,832.0	-11,606.0	-11,601.4	-11,269.0
	LR1_1_DR1_4_VC3_V12b	-40,025.8	-23,250.7	-11,287.0	-10,578.2	-10,514.3	-10,085.0
	LR1_1_DR1_4_VC3_V8a	-31,827.2	-17,082.7	-6,691.0	-5,795.3	-5,740.5	-5,628.0
	LR1_1_DR1_4_VC3_V9a	-32,275.6	-18,656.8	-9,383.0	-8,190.1	-8,122.7	-7,696.0
	LR1_2_DR1_3_VC2_V6a	-30,560.5	-28,237.6	-15,841.0	-14,498.2	-14,490.9	-13,810.0
	LR1_2_DR1_3_VC3_V8a	-36,734.0	-33,683.6	-17,379.0	-15,414.3	-15,249.4	-14,931.0
	LR2_11_DR2_22_VC3_V6a	-30,989.4	-21,553.1	-14,198.0	-15,066.7	-15,063.1	-13,351.0
	LR2_11_DR2_33_VC4_V11a	-47,031.0	-32,146.6	-19,565.0	-18,735.6	-18,705.2	-17,008.0
	LR2_11_DR2_33_VC5_V12a	-56,561.9	-38,272.3	-25,988.0	-24,785.7	-24,704.3	-24,246.0
	LR2_22_DR2_22_VC3_V10a	-59,324.2	-54,495.2	-35,873.0	-35,455.3	-35,300.2	-34,167.0
	LR2_22_DR3_333_VC4_V14a	-63,355.4	-59,797.5	-33,503.0	-31,139.6	-31,030.4	-29,931.0
	LR2_22_DR3_333_VC4_V17a	-72,785.7	-66,622.3	-33,909.0	-31,316.8	-31,308.8	-30,227.0

Source: From the author (2020).

Table A.2: Metaheuristic results considering instances with $|T| = \{45, 60\}$.

$\|\mathcal{T}\|$	Instance	Objective Value			Time (s)			Gap (BKV)		
		MS	LNS	RMIP	MS	LNS	RMIP	MS	LNS	RMIP
45	LR1_1_DR1_3_VC1_V7a	-13,272	-13,272	-13,272	4.6	244.0	0.1	0.00	0.00	0.00
	LR1_1_DR1_4_VC3_V11a	-10,819	-11,167	-11,202	17.0	801.1	20.8	3.92	0.68	0.37
	LR1_1_DR1_4_VC3_V12a	-10,516	-10,699	-10,732	24.5	954.9	31.3	2.05	0.31	0.00
	LR1_1_DR1_4_VC3_V12b	-8,572	-9,001	-9,001	24.4	1,120.6	14.3	5.99	0.93	0.93
	LR1_1_DR1_4_VC3_V8a	-3,973	-4,934	-5,021	8.8	509.7	4.6	28.51	3.49	1.69
	LR1_1_DR1_4_VC3_V9a	-5,661	-6,540	-6,633	11.1	509.8	30.3	22.25	5.82	4.35
	LR1_2_DR1_3_VC2_V6a	-8,986	-9,422	-9,748	5.6	381.7	52.0	23.90	18.17	14.23
	LR1_2_DR1_3_VC3_V8a	-11,500	-11,894	-11,953	12.7	833.4	37.0	4.44	0.97	0.48
	LR2_11_DR2_22_VC3_V6a	177,987	-	-9,301	4.9	234.6	10.6	-	-	4.57
	LR2_11_DR2_33_VC4_V11a	-11,899	-12,525	-12,879	37.1	485.2	120.0	17.80	11.91	8.90
	LR2_11_DR2_33_VC5_V12a	214,147	-16,690	-17,788	48.3	621.0	120.0	-	10.39	3.59
	LR2_22_DR2_22_VC3_V10a	21,805	-24,241	-24,496	37.1	834.1	120.0	-	2.53	1.47
	LR2_22_DR3_333_VC4_V14a	-18,633	-18,654	-19,525	143.9	375.8	120.0	17.81	17.68	12.84
	LR2_22_DR3_333_VC4_V17a	-18,419	-18,533	-18,946	259.4	610.6	120.0	21.04	20.30	17.73
60	LR1_1_DR1_3_VC1_V7a	-16,606	-16,675	-16,675	7.5	359.2	4.5	0.42	0.00	0.00
	LR1_1_DR1_4_VC3_V11a	-12,397	-12,513	-12,650	40.2	1,183.2	75.2	6.94	5.95	4.81
	LR1_1_DR1_4_VC3_V12a	-10,850	-11,000	-11,026	59.6	1,199.7	80.0	1.75	0.36	0.13
	LR1_1_DR1_4_VC3_V12b	-9,727	-10,015	-10,016	58.5	2,007.2	80.0	3.35	0.38	0.37
	LR1_1_DR1_4_VC3_V8a	133,056	19,397	-4,150	21.3	622.4	80.0	-	-	25.18
	LR1_1_DR1_4_VC3_V9a	-6,522	-6,930	-7,092	26.9	704.2	80.0	15.79	8.97	6.51
	LR1_2_DR1_3_VC2_V6a	-12,488	-12,781	-13,085	12.9	542.3	80.0	9.16	6.65	4.17
	LR1_2_DR1_3_VC3_V8a	-13,706	-14,400	-14,424	26.7	1,322.3	80.0	6.90	1.75	1.59
	LR2_11_DR2_22_VC3_V6a	476,348	163,109	69,589	10.6	405.4	96.3	-	-	-
	LR2_11_DR2_33_VC4_V11a	436	-1,288	-7,883	77.9	238.7	160.0	-	-	11.59
	LR2_11_DR2_33_VC5_V12a	355,857	87,674	5,272	86.7	401.4	160.0	-	-	-
	LR2_22_DR2_22_VC3_V10a	83,612	-23,930	-26,048	61.1	438.5	160.0	-	36.34	6.57
	LR2_22_DR3_333_VC4_V14a	17,600	17,600	2,900	246.0	423.0	160.0	-	-	-
	LR2_22_DR3_333_VC4_V17a	-22,818	-22,818	-23,196	420.9	649.8	160.0	18.33	18.33	16.42

Source: From the author (2020).

Table A.3: Metaheuristic results considering instances with $|T| = \{45, 60\}$ and the highest parameter values tested.

$\|\mathcal{T}\|$	Instance	Objective Value			Time (s)			Gap (BKV)		
		MS	LNS	RMIP	MS	LNS	RMIP	MS	LNS	RMIP
45	LR1_1_DR1_3_VC1_V7a	-13,272	-13,272	-13,272	40.3	2,245.1	0.1	0.00	0.00	0.00
	LR1_1_DR1_4_VC3_V11a	-11,038	-11,101	-11,244	155.8	2,162.8	3,178.6	1.86	1.28	-0.01
	LR1_1_DR1_4_VC3_V12a	-10,540	-10,694	-10,747	224.4	4,516.9	2,366.6	1.83	0.36	-0.14
	LR1_1_DR1_4_VC3_V12b	-8,591	-8,995	-9,031	225.8	4,411.5	2,013.6	5.74	1.01	0.60
	LR1_1_DR1_4_VC3_V8a	-3,973	-5,032	-5,104	79.6	2,734.2	87.4	28.51	1.47	0.03
	LR1_1_DR1_4_VC3_V9a	-5,671	-6,548	-6,853	99.8	1,825.9	352.4	22.04	5.70	1.01
	LR1_2_DR1_3_VC2_V6a	-9,015	-9,524	-9,887	50.8	1,497.2	3,600.0	23.51	16.91	12.62
	LR1_2_DR1_3_VC3_V8a	-11,573	-11,908	-11,984	162.6	5,750.4	3,118.6	3.78	0.86	0.21
	LR2_11_DR2_22_VC3_V6a	177,987	-9,354	-9,498	56.6	1,466.7	22.2	-	3.90	2.32
	LR2_11_DR2_33_VC4_V11a	-12,548	-12,646	-13,061	374.8	1,549.8	3,600.0	11.71	10.86	7.33
	LR2_11_DR2_33_VC5_V12a	175,383	-16,733	-18,225	460.9	3,784.7	3,600.0	-	10.12	1.09
	LR2_22_DR2_22_VC3_V10a	2,323	-24,057	-24,875	349.4	2,330.1	3,600.0	-	3.32	-0.08
	LR2_22_DR3_333_VC4_V14a	-19,529	-19,551	-20,744	1,394.5	3,417.0	3,600.0	12.43	12.30	5.86
	LR2_22_DR3_333_VC4_V17a	-18,824	-19,005	-19,944	2,582.5	7,877.8	3,600.0	18.44	17.32	11.82
60	LR1_1_DR1_3_VC1_V7a	-16,610	-16,675	-16,675	74.6	3,738.8	774.7	0.39	0.00	0.00
	LR1_1_DR1_4_VC3_V11a	-12,514	-12,517	-13,052	281.8	1,794.5	3,600.0	5.94	5.91	1.60
	LR1_1_DR1_4_VC3_V12a	-10,924	-11,020	-11,116	392.5	3,739.1	3,600.0	1.06	0.19	-0.69
	LR1_1_DR1_4_VC3_V12b	-9,874	-9,938	-10,006	377.1	3,737.6	3,600.0	1.82	1.16	0.47
	LR1_1_DR1_4_VC3_V8a	131,765	2,598	-4,730	165.7	2,328.0	3,600.0	-	-	10.02
	LR1_1_DR1_4_VC3_V9a	-6,678	-7,084	-7,404	327.0	2,744.4	3,600.0	13.09	6.64	2.00
	LR1_2_DR1_3_VC2_V6a	-12,545	-12,875	-13,271	172.1	3,267.5	3,600.0	8.66	5.88	2.72
	LR1_2_DR1_3_VC3_V8a	-13,938	-14,350	-14,526	325.5	5,012.4	3,600.0	5.12	2.11	0.87
	LR2_11_DR2_22_VC3_V6a	462,475	34,926	25,128	155.3	3,250.1	3,600.0	-	-	-
	LR2_11_DR2_33_VC4_V11a	-14,238	-14,240	-14,608	1,072.6	2,523.5	3,600.0	8.09	8.08	5.38
	LR2_11_DR2_33_VC5_V12a	278,830	39,981	-22,039	1,243.4	5,212.1	3,600.0	-	-	4.14
	LR2_22_DR2_22_VC3_V10a	36,327	-26,984	-30,833	1,064.7	3,433.7	3,600.0	-	11.22	5.88
	LR2_22_DR3_333_VC4_V14a	-7,452	-7,653	-16,430	2,296.1	4,858.7	3,600.0	-	-	14.13
	LR2_22_DR3_333_VC4_V17a	-23,349	-23,424	-23,717	3,873.9	8,702.1	3,600.0	15.64	15.28	13.86

Source: from the author (2020).

Table A.4: Metaheuristic results for the long planning horizon instances (considering only the instances with feasible solutions).

$\|\mathcal{T}\|$	Instance	Objective Value			Time (s)			Gap (LB)		
		MS	LNS	RMIP	MS	LNS	RMIP	MS	LNS	RMIP
120	LR1_1_DR1_3_VC1_V7a	-29,929	-30,184	-30,198	35.2	1,220.2	160.0	4.02	3.14	3.09
	LR1_1_DR1_4_VC3_V11a	-18,375	-18,375	-18,602	147.8	376.9	160.0	15.07	15.07	13.69
	LR1_1_DR1_4_VC3_V12a	-15,287	-15,360	-15,503	197.3	624.6	160.0	17.72	17.18	16.10
	LR1_1_DR1_4_VC3_V12b	-12,147	-12,234	-12,263	196.3	877.7	160.0	18.48	17.64	17.36
	LR1_1_DR1_4_VC3_V9a	-7,771	-7,824	-8,038	97.6	367.0	160.0	31.84	30.97	27.46
	LR1_2_DR1_3_VC2_V6a	-21,287	-21,295	-21,785	47.98	551.59	160.00	19.69	19.64	17.01
	LR1_2_DR1_3_VC3_V8a	-22,796	-22,905	-22,959	89.7	393.6	160.0	11.80	11.28	11.02
180	LR1_1_DR1_3_VC1_V7a	-43,041	-43,144	-43,223	97.8	901.1	240.0	3.57	3.32	3.13
	LR1_1_DR1_4_VC3_V11a	-23,998	-23,998	-24,000	341.2	698.1	240.0	17.05	17.05	17.04
	LR1_1_DR1_4_VC3_V12b	-14,399	-14,496	-14,501	474.6	702.3	240.2	21.79	20.99	20.95
	LR1_1_DR1_4_VC3_V9a	-8,870	-8,870	-9,478	230.3	508.5	240.0	18.10	18.10	17.43
	LR1_2_DR1_3_VC2_V6a	-30,830	-30,830	-31,013	121.9	566.3	240.0	18.10	18.10	17.43
	LR1_2_DR1_3_VC3_V8a	-31,497	-31,497	-31,652	222.9	643.2	240.0	14.94	14.94	14.38
360	LR1_1_DR1_3_VC1_V7a	-81,675	-81,675	-81,815	646.1	986.6	480.0	3.78	3.78	3.60
	LR1_1_DR1_4_VC3_V11a	-39,489	-39,489	-39,625	2,013.7	2,989.0	480.0	23.30	23.30	22.86

Source: From the author (2020).

APPENDIX B — MÉTODOS DE SOLUÇÃO PARA UM PROBLEMA DE ROTEAMENTO DE INVENTÁRIO MARÍTIMO

O transporte marítimo é o meio de transporte mais utilizado quando considerados o transporte de grandes volumes. De acordo com UNCTAD (2019), o total transportado por esse meio foi de aproximadamente 11 bilhões de toneladas, um valor que vem crescendo ao longo de vários anos. O mesmo vem ocorrendo com a frota mundial de navios, que cresceu 2,6% entre o início de 2018 e o início de 2019. Tais perspectivas de crescimento do setor levam a problemas mais desafiadores considerando frotas de navios, cargas, e gerenciamento de estoque. O uso de estratégias de otimização acaba sendo crítica para esse setor de capital intensivo, onde uma pequena melhora nas operações podem levar a uma economia significante de recursos (AGRA et al., 2013).

Esse trabalho explora métodos de solução para o Problema de Roteamento de Inventário Marítimo (MIRP - *Maritime Inventory Routing Problem*), um problema desafiador que envolve decisões relacionadas a definição de rotas e escalonamentos de navios em portos de carga e descarga, bem como definir a quantidade de um produto que deve ser carregada ou descarregada em cada visita de a um determinado porto, de modo que os estoques de tais produtos fiquem entre limites inferiores e superiores ao longo de um horizonte de planejamento.

Uma extensa revisão da literatura baseada no problema foi apresentada apresentando duas tabelas que resumem o conteúdo dos trabalhos analisados, classificando-os em diferentes grupos conforme a característica do PRIM estudado em cada trabalho.

O primeiro método proposto consiste em uma estrutura matheuristica, que combina a utilização de técnicas heurísticas com programação linear inteira. A estrutura é composta por dois métodos principais: o algoritmo *relax-and-fix* (R&F), e o algoritmo *fix-and-optimize* (F&O). O algoritmo R&F é responsável por construir uma solução inicial para o problema. Isso é feito dividindo-se o problema em subproblemas menores através da relaxação de parte das variáveis inteiras do modelo original. Iterativamente o algoritmo resolve os subproblemas e fixa as variáveis inteiras nos valores encontrados em iterações anteriores. Já o algoritmo F&O é responsável por melhorar as soluções encontradas pelo R&F. Ele consiste em inicialmente fixar os valores das variáveis inteiras e iterativamente permitir que parte dessas variáveis possam ser otimizadas (não-fixadas) por um resolvedor matemático. A estratégia na qual as variáveis a serem otimizadas são selecionadas define uma vizinhança na solução. Neste trabalho foram utilizadas quatro estratégias, três baseadas na literatura e uma originalmente proposta neste trabalho.

Foram realizados testes com a estrutura matheuristica considerando duas formulações do problema: a formulação original que consiste em uma formulação de tempo-espaço, proposta por (PAPAGEORGIOU et al., 2014b), e uma formulação de fluxo de carga fixa, proposta por (AGRA et al., 2013) e adaptada para a variante do PRIM estudado neste trabalho. Junto às formulações foram propostas diferentes melhorias como a utilização de desigualdades válidas, restrições adicionais e pré-processamento.

O segundo método de solução proposto para o MIRP foi uma estrutura metaheuristica composta por um algoritmo *multi-start*, e por um algoritmo *large neighborhood search*. O algoritmo multi-start é responsável por construir soluções iniciais ao problema. Para esse algoritmo foi proposta uma heurística gulosa específica para o MIRP, a qual possui componentes aleatórios para que sejam criadas várias soluções que explorem diferentes espaços de busca do problema. Uma das maiores vantagens dessa abordagem é a sua rapidez na construção de soluções se comparada com um método que necessita de um resolvedor matemático. O algoritmo large neighborhood search seleciona um subconjunto das soluções criadas pelo algoritmo de muitos-inícios e tenta melhorá-las a partir da destruição parcial da solução e reconstrução dela de uma maneira diferente. O algoritmo seleciona aleatoriamente um conjunto de navios para destruir as suas rotas e posteriormente usa a heurística gulosa do algoritmo de muito inícios para reconstruir a rota desses navios de uma maneira diferente. Ainda, foi proposto um elemento de pós-melhoria do método, o qual consiste em criar um modelo matemático reduzido considerando apenas as variáveis que obtiveram um valor positivo correspondente na solução metaheuristica, e após resolver o modelo reduzido com um resolvedor matemático.

Os testes computacionais foram realizados com instâncias de uma biblioteca pública específica para o MIRP. Em ambos os métodos foram feitos testes com a parametrização dos algoritmos utilizando valores definidos a priori e também utilizando a ferramenta de configuração automática de parâmetros *irace*. Para a estrutura matheuristica foi avaliado a contribuição de cada melhoria proposta para as formulações considerando a qualidade da solução e o tempo de processamento Foi observado que a formulação de fluxo de carga fixa possui uma performance superior ao modelo de tempo-espaço, tanto na obtenção de soluções de melhor qualidade quanto no número de soluções factíveis encontradas e no tempo de processamento. Já para o método metaheurístico os resultados demonstraram a potencialidade do algoritmo em resolver instâncias do problema em um tempo de processamento menor. Com isso foi possível prover soluções para instâncias do problema com longos horizontes de planejamento, as quais pelo nosso conhecimento, não houveram tentativas de solução reportadas na literatura.

Ambos métodos de solução propostos demonstraram ser promissores para prover soluções para o problema, sendo capazes de obter novas melhores soluções conhecidas para as instâncias testadas.

A natureza do problema induz a uma grande dificuldade em obter soluções factíveis para o problema, sendo que isso foi observado em nossos métodos de solução. Além disso, pode-se observar que a definição dos valores do parâmetros dos algoritmos é muito dependente da instância testada. Com isso, uma melhoria a ser considerada provém da utilização de ferramentas de seleção algorítmica para definir o melhor conjunto de parâmetro para cada instância a partir de suas características. Tal abordagem utiliza técnicas de aprendizado de máquina para definir o melhor conjunto de parâmetros. Um desafio a ser enfrentado nessa abordagem é a obtenção de um grande conjunto de instâncias, que infelizmente ainda é limitado considerando instâncias públicas de teste. Outro tópico de pesquisa consiste em utilizar técnicas mais aprimoradas para a abordagem metaheurística, como a adoção de mecanismos de memória no algoritmo multi-inícios para guiar a construção de soluções, bem como a adoção de técnicas de busca em vizinhança grande adaptativa, onde técnicas específicas (e não aleatórias) são utilizadas para direcionar a busca.