

Article

Ship Routing with Pickup and Delivery for a Maritime Oil Transportation System: MIP Model and Heuristics

Vinicius P. Rodrigues ^{1,2,*}, Reinaldo Morabito ^{2,*}, Denise Yamashita ², Bruno J. V. da Silva ² and Paulo C. Ribas ³

¹ Section of Engineering Design and Product Development, Department of Mechanical Engineering, Technical University of Denmark (DTU), Produktionstorvet, Kongens Lyngby 2800, Denmark

² Production Engineering Department, Federal University of São Carlos (UFSCar), Via Washington Luiz, km 235, São Carlos/SP 13565-905, Brazil; denisesy@dep.ufscar.br (D.Y.); bruno.silva@ufscar.br (B.J.V.d.S.)

³ PETROBRAS, Research and Development Center (CENPES), Av. Horácio de Macedo, 950, Ilha do Fundão, Rio de Janeiro/RJ 21941-598, Brazil; paulo.ribas@petrobras.com.br

* Correspondence: vipiro@dtu.dk (V.P.R.); morabito@ufscar.br (R.M.); Tel.: +55-3351-9516 (R.M.)

Academic Editor: Ockie Bosch

Received: 13 July 2016; Accepted: 12 September 2016; Published: 21 September 2016

Abstract: This paper examines a ship routing problem with pickup and delivery and time windows for maritime oil transportation, motivated by the production and logistics activities of an oil company operating in the Brazilian coast. The transportation costs from offshore platforms to coastal terminals are an important issue in the search for operational excellence in the oil industry, involving operations that demand agile and effective decision support systems. This paper presents an optimization approach to address this problem, based on a mixed integer programming (MIP) model and a novel and exploratory application of two tailor-made MIP heuristics, based on relax-and-fix and time decomposition procedures. The model minimizes fuel costs of a heterogeneous fleet of oil tankers and costs related to freighting contracts. The model also considers company-specific constraints for offshore oil transportation. Computational experiments based on the mathematical models and the related MIP heuristics are presented for a set of real data provided by the company, which confirm the potential of optimization-based methods to find good solutions for problems of moderate sizes.

Keywords: pickup and delivery with time windows; maritime transportation; oil industry; relax-and-fix heuristics; time decomposition heuristics

1. Introduction

Maritime transportation plays an important role in moving large volumes of cargo over large distances, such as cabotage in certain countries or import and export operations. One of the key economic activities in the Brazilian coast is oil exploration. The country currently presents a daily production of approximately 2.7 million barrels, with around 80% of this total volume being explored in offshore mode [1,2]. An important driver for the Brazilian oil industry is the current exploration of the pre-salt layer, which is a region of rocks located in deep waters of the Brazilian coast and presents significant potential for oil exploration. This depth can reach over 7000 m; the term “pre” is used because these rocks were deposited before the salt layer of the ocean floor. Due to the commercialization boost of oil products and the discovery of new sources for exploration, oil companies in Brazil are increasingly facing pressure for cost reduction. Therefore, oil companies are seeking higher levels of agility and efficiency in their decision-making processes, as well as developing strategies to better organize their activities for enhanced economic outcomes, especially those activities related to transportation operations of commodities [3–5]. Within this context, the oil sector requires versatile

operations from the refineries that are receiving oil to be processed, and this, in turn, is reflected in larger volumes and tight deadline requirements with which logistics are to comply [6,7].

In this paper, we examine a routing and scheduling problem for oil transportation ships from offshore platforms to coastal terminals, mainly driven by operations of an oil company operating in Brazil. An optimization approach based on mixed integer programming is presented to properly represent the oil pickup at the platforms and delivery at the terminals in the context of this company's operation. Relax-and-fix and time decomposition heuristics are also presented to solve the problem using the GAMS/CPLEX optimization package, in order to capture the fundamental business requirements and specific characteristic of the company's highly dynamic environment. Furthermore, we address a problem that falls into the category of industrial operating mode [8,9]. Some recent applications of emerging solution methods have been also proposed for tramp operating mode [10,11] and liner operating mode [12,13]. It should be noted that although there are other related works in the literature in this line of research, we are not aware of other studies exploring models and solution methods addressing this specific maritime oil pickup and delivery problem with all its particular operational characteristics.

This study was conducted in close collaboration with an oil company operating on the Brazilian coast, so that the problem could be well characterized and validated in terms of market environment and fundamental characteristics of the current operations. The proposed model captures many important elements of this real system. However, it is difficult to obtain optimal solutions considering the size of the real problem instances of the company, a fact that also motivated the development of specific MIP-heuristic procedures based on relax-and-fix and time decomposition approaches. This paper is structured as follows: Section 2 highlights studies from the relevant literature on pick and delivery problems, as well as applications of MIP heuristics based on relax-and-fix and time decompositions; Section 3 describes the problem studied; Section 4 presents the MIP model to represent the problem; Section 5 describes the heuristic methods based on relax-and-fix and time decomposition; and Section 6 presents and analyzes the computational results. Finally, some concluding remarks and future research prospects are presented and discussed in Section 7.

2. Literature Review

The general pickup and delivery problem (PDP) is based on the idea that a set of routes is to be built in order to meet specific transportations requests [14]. The nodes in the graph are divided into either pickup or delivery nodes, and a fleet of vehicles is available. A pickup node poses a specific demand, which must be collected and respectively delivered to the corresponding delivery node; therefore, in this problem the number of pickups must be equal to the number of deliveries. Additionally, the pickup nodes are to be visited before the delivery node, and the pickup-delivery pair must be positioned in the same route [15,16]. The PDP variant that considers time windows, namely the pickup and delivery problem with time windows (PDPTW), is also widely described in the literature (e.g., [16,17]). This problem has allowable pickup/delivery time frames assigned to nodes. The visit at a node is allowed to begin within a time window defined between the earliest and latest time specified for visiting the node [16]. Furthermore, the works of [7,15,18–23] provide insightful details on multiple PDP formulations and applications.

Several heuristics have also been proposed to tackle different variants of PDP problems. For instance, Bent et al. [24] proposes a two-stage heuristic composed of simulated annealing in the first stage and large neighborhood search in the second stage with a view toward minimizing total cost. Additionally, Mitrović-Minić et al. [25] addresses the dynamic PDPTW, in which not all of the requests are known in advance, by proposing a double-horizon (short- and long-term considerations) heuristic in combination with insertion and improvement heuristics. The work carried forth by [26] proposes an extension of the large neighborhood search (LNS) that had been traditionally suggested in the literature for solving vehicle routing problems with time windows; the heuristic framework is named the adaptive large neighborhood search (ALNS) and is composed of multiple competing

sub-heuristics, which are selected according to previous performance. Tabu search has also been explored in the literature. For instance, Nanry et al. [27] suggest a reactive tabu search procedure focused on the typical precedence and coupling constraints of the PDPTW.

In the last few years, many studies have also developed MIP-based heuristics to solve complex problems in optimization. The idea is to decompose the main problem and generate sub-problems that are easier to solve, relaxing complicating constraints or integrality restrictions. A widely-known MIP heuristic is the relax-and-fix heuristic described in [28,29]. Several studies that propose a relax-and-fix heuristic to solve optimization problems can be found in the literature. For example, relax-and-fix-based heuristics for solving lot sizing and scheduling problems were proposed in [28,30–34]. The study [35] applied a relax-and-fix based heuristic to solve the travelling umpire problem, whereas Noor-E-Alam et al. [36] applied the method to solve large-scale grid-based location, and Kjeldsen et al. [37] optimize the production plan for power-generating companies.

With respect to maritime transportation problems, the first attempts towards building good solutions were based on linear optimization models [38,39]. Since then, a number of exact solution methods and heuristics were proposed; some recent reviews can be found in [11,40,41]. Among the reviewed works of the literature, we highlight the study carried out by [42], which presented a maritime inventory-routing problem for ammonia pickup and delivery with considerations of time windows, predefined routes, inventory control and the minimization of shipping cost. The work of [42] and the references therein provide different views on the general modeling of ship routing problems, which served as a basis for this work.

Furthermore, Al-Khayyal et al. [43] also studied a maritime inventory-routing problem with heterogeneous fleet and routing cargo, where the ship's compartments and the decisions of the quantities to be loaded/unloaded were also explicitly considered. The work of [44] dealt with an allocation problem of oil in the context of a Brazilian oil company, which determines boarding plans for tankers to transport oil from platforms to terminals, in addition to the product allocation plans, which positions the study within an aggregate planning level (tactical), without detailed consideration of the operations of ship routing and scheduling. Additionally, [7] studied a routing problem in oil maritime transportation with load separation, which employed a specific procedure for pre-generating the routes for the tankers to follow.

The LNS heuristic was also applied to maritime transportation by [45] with considerations of split loads for tramp shipping companies. Additionally, Bakkehaug et al. [46] applies the ANLS heuristic to a maritime setting, with a focus on fleet deployment and voyage (i.e., sailing on a route) separation. A multi-start local search heuristic was proposed by [47] for short-term ship scheduling problems in the tramp mode. The tabu search approach derived by [48] for ship routing and scheduling problems is tested and benchmarked against the multi-start heuristic [47]; the tabu search heuristic yields better solutions for large and tightly-constrained instances. Another tabu search method, considering flexible cargo quantities, is proposed and presented by [49]. On the domain of dynamic and stochastic problems, Tirado et al. [50] proposes three heuristic procedures—myopic dynamic heuristic, multiple scenario with consensus and branch-and-regret—for PDP with dynamic requests and stochastic processes for the arrival of new cargoes. Additionally, Agra et al. [51] proposes a stochastic approach for maritime inventory routing with uncertainty in sailing and port service times at the archipelago of Cape Verde.

In this paper, we propose two MIP heuristics. The first one is a relax-and-fix heuristic, and the second one is a time-based decomposition heuristic. Although these types of heuristics have been consistently applied towards tackling a number of problems in production planning and control, as previously described in this section, we emphasize that the only study found in the literature that applied a relax-and-fix method to a routing problem was presented by [52]. In [52], the authors propose a modification of the relax-and-fix heuristic and apply it to solve the inventory routing problem. First, they propose a relax-and-fix heuristic based on time decomposition in order to construct an initial feasible solution. In order to reduce the effort from repeatedly solving many continuous variables at the end of the model horizon, the authors propose adopting a simplified linear model or completely omitting this part of the model horizon. After an initial feasible solution is obtained, an improvement

solution heuristic, also based on relax-and-fix, is applied. The improvement method consists of dividing the time horizon into intervals and iteratively removing the fixing of the integer variables in this interval, then re-optimizing. The computational tests show that the proposed method presents good quality solutions and a reduction in computational time.

While relax-and-fix-based heuristics are successful when applied to many different optimization problems, the large size of some real-world problems may present a challenge even to the relaxed model. In this paper, we present a second decomposition-based heuristic where, at each iteration, just a subset of the variables is included in the model, while the remaining variables are omitted. In the next iteration, the variables considered in the previous iteration are fixed, and a new subset of variables is included in the model. This kind of strategy to reduce the number of variables was also used in [35,52].

3. Characterization of the Problem

Some of the fundamental characteristics of the particular problem examined in this paper include a heterogeneous ship fleet and highly-scattered offshore platforms and terminals along the coast. One of the main issues faced by maritime transportation operations of offshore oil exploration is the scheduling of ships to meet a set of move requests with origins, destinations and pre-established amounts of products, defined by the logistics tactical planning of the company. Pickups at the offshore platforms and deliveries at coastal (onshore) terminals should be performed within predefined time windows over a planning horizon (i.e., several days to a few weeks in this particular case). Both platforms and terminals are called “operating sites”.

It is important to highlight that a ship is allowed to perform more than one ordered pickup on the same visit to a platform, as well as more than one ordered delivery on the same visit to a terminal, depending on the respective pickup and delivery time windows and the available capacity of the ship. Each ship may be in a different location at the start of the planning horizon (with known location in advance) and may end in a different location at the end of such a horizon (with either unknown or known location in advance, the latter in the case of scheduled maintenance or other operation for the ship). Most ships are available during the whole planning horizon; however, there may be ships available only after the beginning of the horizon. In practice, the problem decisions are taken in a rolling time horizon basis, in which a ship routing and scheduling plan is determined for the whole planning horizon, but only the decisions in initial time periods are implemented to protect the operation against uncertainties in the following periods.

An important aspect of the problem is the consideration of oil inventory both in the platforms and terminals, which is dealt with implicitly through the use of time windows. These time windows are defined on a more aggregated planning level, in which the variations and limitations of oil inventory are considered. Platforms hold their own inventory, which may not exceed a specified maximum operating limit (popularly called “top” by the operators), since it results in a halt of platform production and incurs prohibitive opportunity costs. This situation arises as a major concern for the company’s maritime logistics decision-makers. For example, it is assumed that the tops of the platforms are directly used to determine the time windows, as well as minimum oil inventory levels for the platforms. This minimum amount of oil is required to ensure physical stability in the platforms for safe operations. The minimum oil inventory acts as a ballast for the platform. The openings of the time windows are calculated based on the predefined size of the batch to be collected, the storage capacity and the specific platform’s production rate. Figure 1 shows an illustration of a graph with the inventory levels along time. The time window opens when the oil inventory level in the platform is large enough to fulfill the demand of the order, and it closes 48 h before it reaches the maximum inventory level capacity, as a strict safety requirement.

Based on the pickup time window, the respective time window for delivery is calculated by adding to the opening of the pickup time window: (i) 48 h of service time in the platform; and (ii) the estimated travel time for the ship to arrive at the terminal. This duration of 48 h for service time was suggested by company experts, and it includes all time spent in docking operations, such as loading time and

any additional waiting time. Similarly, the end of the time window for delivery in the terminal is determined by adding to the end of the pickup time window: (i) 48 h; and (ii) the estimated travel time for the ship to travel from the platform to the terminal.

The objective is to minimize costs related to fuel consumption and freighting contracts. There are also berthing costs associated with the operating sites; these costs stimulate multiple ordered pickups (or deliveries) to be performed on the same visit to a platform (or terminal), if the ship's available capacity and time windows of these pickups (or deliveries) allow. Since the visit to a platform is costly and consecutive visits to different platforms pose additional risks to the berthing operations on the subsequent platform(s), a penalty is also imposed for consecutive visits at different platforms in the same itinerary. The increased risks emerge from the fact that, in consecutive visits to different platforms, the ship performs subsequent visits with oil loads from previous visits already onboard. The ideal situation for the company's operation would be to encompass a series of sequential platform-terminal visits for the ships, instead of making sequential visits to different platforms. However, this ideal type of ship route might render the problem infeasible in many practical circumstances; therefore, it is important to penalize consecutive platform visits as a way to build feasible solutions while avoiding higher costs.

The problem can be represented as a pickup and delivery vehicle routing with time windows, heterogeneous fleet, multiple depots (initial locations of each ship) and multiple visits to each platform and terminal throughout the planning horizon, accompanied by additional constraints that include specificities of the operation in this study. As a starting point for modeling this problem, a well-known formulation for the pickup and delivery problem with time windows [53] was chosen.

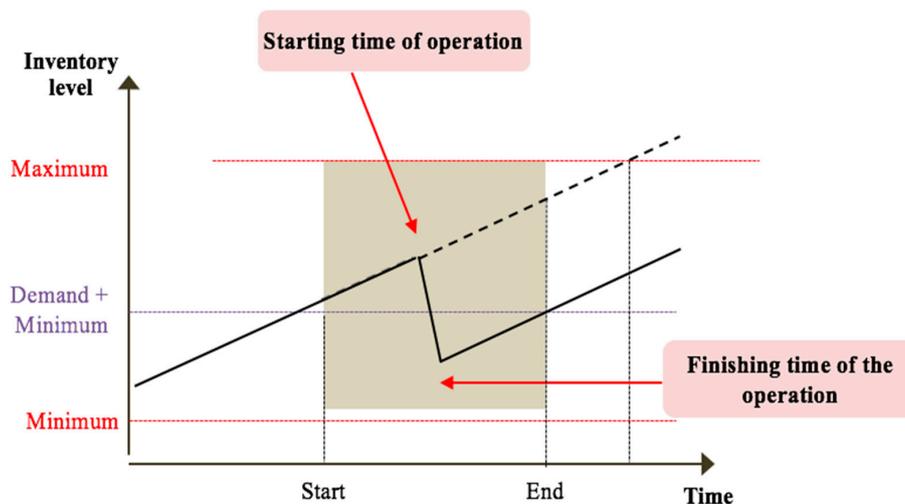


Figure 1. Time window computation.

It is important to highlight that since the origins and destinations are predefined by each origin-destination pair in the more aggregate system planning of the company, the ordered quantities are therefore input data for the present problem. The maritime transportation of oil derivatives (such as gasoline, diesel fuel, jet fuel, etc.) is beyond the scope of this paper. Additionally, the fleet of oil tankers is highly heterogeneous, with different operating costs, maximum and minimum speeds and capabilities. Furthermore, ships feature three additional characteristics that define their ability to berth in certain operating sites: (i) ship draft; (ii) the length or LOA (length overall); and (iii) dynamic positioning devices.

The draft is defined as the vertical distance, taken on a transversal plane, between the lower end of the ship and the line determined by the intersection of the surface of the water with the outer surface of the hull. The draft varies as a function of the load onboard of the ship [54]; therefore, it is possible to limit the loads as a response to draft constraints that might occur in practice (for further

details on problems addressing draft issues and constraints, see, e.g., [55–57]). The draft constraints are highly dependent on the terminal's access channels and the berth's characteristics. Additionally, LOA, also known as “length of wheel to wheel”, refers to the distance between the salient points of the front and rear of the ship, meaning its total length. The LOA and the draft of the ship may impose constraints on its approach to a terminal. Figure 2 shows an illustration of these two features.

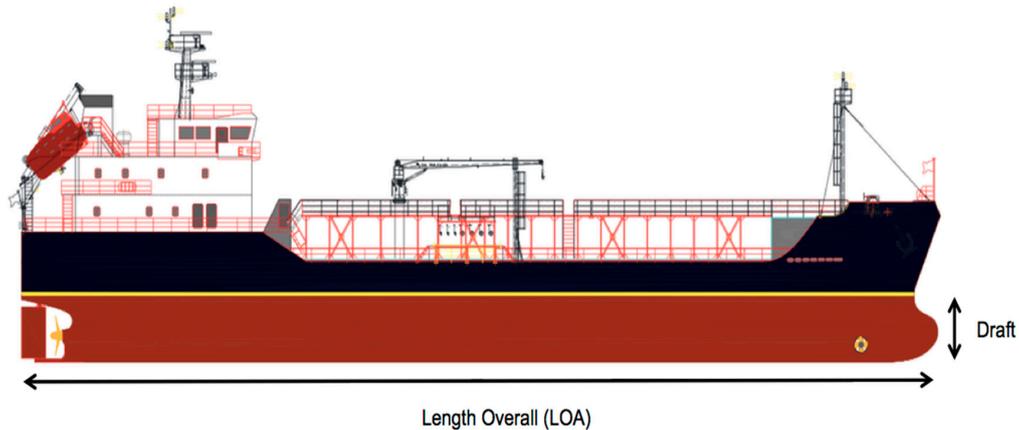


Figure 2. Illustration of an oil tanker with the respective markings of LOA and draft. Source: adapted from [58].

Some ships also present a system called dynamic positioning (DP), which, according to [59], maintains floating structures in a fixed position or pre-defined tracks for marine operation purposes through the use of active thrusters. About 50 different Brazilian platforms allow berthing of all ships with dynamic positioning, but only a few platforms allow berthing of conventional ships. Additionally, there are about 10 terminals along the Brazilian coast, with about 20 berths in total, distributed heterogeneously across the terminals. The berths are specific locations in the terminals, where ships dock in order to perform cargo loading and unloading. Each one of the berths presents physical constraints from draft and LOA that should be respected so that the ships are allowed to berth, since each ship occupies one single berth. However, in practice, as the draft constraint depends on the characteristics of the access channels and the berths at the terminals, this is done by limiting the load onboard the ship to a value lower than the maximum capacity of the ship before berthing at each terminal.

Since the pairs of pickup and delivery are pre-established by the tactical planning of the company, the routes and schedules of the ships must be determined, and they are therefore configured as the main decisions for the problem to support. Most importantly, besides allowing one or more pickups and deliveries in each ship visit to an operating site, this pickup and delivery problem also differs from most other industrial cases regarding the typical distances involved. In most cases of oil exploration around the world, the transportation occurs from several producing companies to several refineries with different rules for responsibility for the freight, typically dealing with very large distances. In Brazil's specific case, the same company produces, refines, plans and operates the transportation, which considerably increases the potential of gains in the logistics operations. The offshore oil explorations in both the North Sea and the Gulf of Mexico bear reasonable resemblance to the Brazilian case.

4. Mathematical Model

The formulation presented in this paper uses a known formulation for the pickup and delivery problem with time windows presented in [53] as a starting point, as well as specific aspects of the present problem derived from the company's business rules. As it is common to have multiple origin-destination pairs scheduled involving the same platform or terminal depending on the length

of the planning horizon, the ship collects, at each visit, the ordered amount that was previously programmed in the batch at the platform and delivers this amount to the corresponding terminal of the origin-destination pair. The problem is represented by a graph, and for modeling purposes, each one of the pickups and deliveries is represented by different nodes, even when the pickups and deliveries are made on the exact same operating site. Therefore, although two nodes in the graph may represent the same operating site, they refer to distinct ship visits for performing one or more ordered pickups or deliveries on each visit.

Figure 3 shows a schematic representation in which the red nodes represent artificial depots (initial and final locations) of the ship, while the black nodes refer to real visits to operating sites. The artificial nodes also have time windows, and the location of the initial artificial depot is defined by the user. The opening of the time window for the initial artificial depot does not necessarily have to correspond to the beginning of the planning horizon; instead, it actually corresponds to the moment at which the ship becomes available for operation. Similarly, the location of the final artificial depot is also defined by the user, in case the ship has a scheduled maintenance procedure or other operations to be performed (e.g., transportation of gasoline, diesel fuel, jet fuel, etc.). The beginning of these other operations corresponds to the end of the time window for the final artificial depot, which, similarly, does not have to coincide with the end of the planning horizon. Otherwise, if not intendedly defined by the user, the location of the final artificial depot is assumed to be the same as the last visit of the ship's route defined by the model solution; in this case, the distances between the operating sites and the final artificial depot are simply defined by the user as zero.

It is possible to check the strategy of duplicating the nodes for the scheduled visits in Figure 3. In this example, a ship leaves the origin artificial depot at Node 1, performs a pickup on a platform at Node 2 and delivers it to a terminal at Node 3. Later on, the ship visits the same platform for another pickup, now represented by a new node (Node 4), to deliver the product at the terminal of Node 5 and finishes its route in the artificial depot, represented as Node 6. Notice that Nodes 2 and 4 refer to the same operating site, but designate different pickups within the planning horizon.

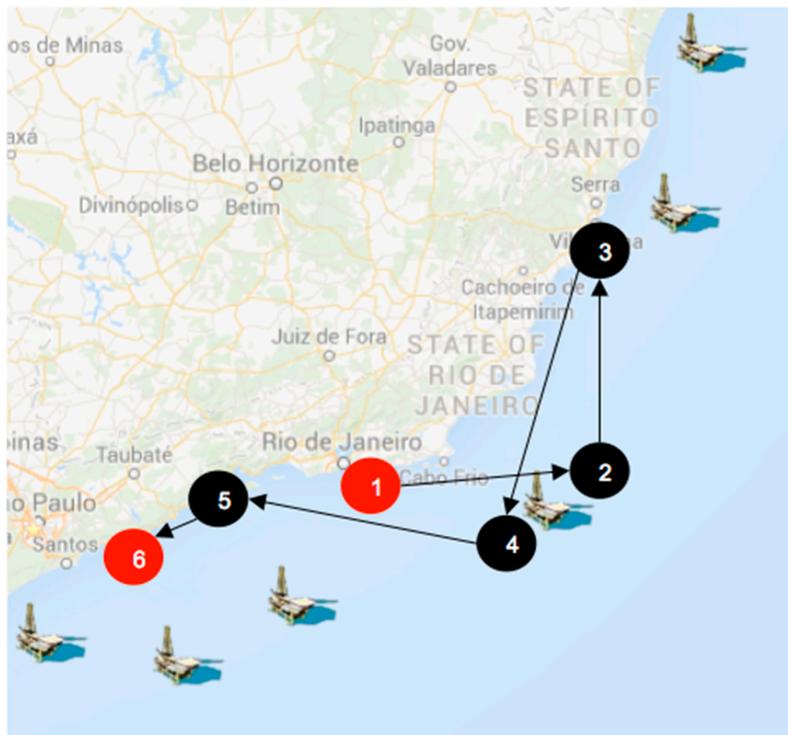


Figure 3. Schematic representation of the replication strategy of nodes and artificial depots adopted in the problem modeling. Map data ©2015 Google.

It is also noteworthy that the illustrations of the initial and final artificial depots in Figure 3 exhibit an arbitrary placement only for the purpose of diagrammatic representation. In practice, the ships usually begin and end their work shifts either near terminals or platforms, in areas known as “anchoring zones”, in which idle ships wait for new requests. However, the modeling approach applied in this paper also considers situations in which some ships may lie outside of the anchoring zones at the beginning of the planning horizon, depending on the characteristics of the requests. It should also be pointed out that the ships travel by routes over the planning horizon, instead of itineraries, since they do not have to necessarily start and finish at the exact same site (artificial depot). With that, we can proceed to presenting the mathematical formulation of the proposed model.

The model parameters are:

Indices and sets:

k is an index that refers to ships ($k \in K$). The total number of ships is $|K|$.

i, j are indices referring to both operating sites (platforms or terminals) and depots.

n is the total number of pickup and delivery pairs (origin-destination pairs).

$O^P = \{1, 2, \dots, n\}$ is the set of nodes where pickups are made (origins).

$O^D = \{n+1, n+2, \dots, 2n\}$ is the set of nodes where deliveries are made (destinations); nodes in O^D are labeled as follows: for each $i \in O^P$, there is a matching node $(i+n)$, wherein node $i \in O^P$ refers to the origin and node $(i+n) \in O^D$ refers to its destination $(i+n)$. Therefore, $|O^P| = |O^D|$.

$O = O^P \cup O^D$ is the total set of nodes that represent platforms and terminals, $|O| = 2n$.

$S = \{s_1, s_2, \dots, s_k\}$ is the set of artificial nodes that indicate the starting node s_k of ship k .

$E = \{e_1, e_2, \dots, e_k\}$ is the set of artificial nodes that indicate the final node e_k of ship k .

$N = O \cup S \cup E$ represents all nodes, with $|N| = 2n + 2|K|$.

$A_{ik}, \forall i \in O, \forall k \in K$ is a 0–1 matrix that indicates whether the ship k cannot berth at node i (equal to one), either due to draft, LOA or other reasons, or if it can berth (equal to zero).

O^{DP} is the set of platforms that only accepts both dynamic positioning ships.

K^{DP} is the set of ships k that feature dynamic positioning.

O_{ik}^{FLEX} is the set of pairs (i, k) , that allow draft flexibility for berthing of ship k .

Parameters:

v_k is the average speed of ship k , expressed in knots.

D_{ij}^{ist} is the distance, expressed in nautical miles, between node $i \in (S \cup O)$ and node $j \in (O \cup E)$. For representing consecutive pickups or deliveries that are performed in the same operating sites (either in a platform or terminal), the corresponding distance is set to zero (i.e., $D_{ij}^{ist} = 0$).

t_{ik}^s is the service time, in hours, at node $i \in O$ serviced by ship k .

$[a_i, b_i]$ is the time window, in hours, on the start of service at node $i \in N$, in which a_i represents its lower bound and b_i the upper bound.

d_i is the load amount (demand), in m^3 , to be picked up or delivered at node $i \in O$, which may have a positive or negative sign. If the demand is positive, it indicates that in operating site i , d_i units of oil must be picked up. If this demand is negative, it indicates that in operating site i , d_i units of product must be delivered. Note that by convention, $d_{i+n} = -d_i$.

C_k^{ap} is the capacity of ship k , in m^3 .

C_{ijk}^s is the navigation (sailing) cost of ship k between node i and j (\$).

C_j^D is the fixed cost of berthing in platform or terminal j , in monetary units. If the ship performs consecutive pickups on the same platform j , the fixed cost is paid only once, when the ship is berthed.

β is a penalty imposed to consecutive visits of the same ship on different platforms in its itinerary, which is undesirable for the company and should, if possible, be avoided.

α_{jk} is the percentage of the maximum load allowed for ship k to berth at an operating site j . This value changes for a ship k berthing at a terminal $j \in O_{jk}^{FLEX}$; for a ship k with DP, $k \in K^{DP}$, berthing at a conventional platform $j \notin O^{DP}$; for a ship k with DP, $k \in K^{DP}$, berthing on a platform that accepts only ships with DP, $j \in O^{DP}$; and for a conventional ship k , $k \notin K^{DP}$, berthing at a platform that accepts conventional ship, $j \notin O^{DP}$.

M is a sufficiently large positive integer number used in the linearization of some constraints.

Variables:

x_{ijk} is a binary variable that takes a value equal to one if ship k runs through arc (i, j) , $i \in (S \cup O)$, $j \in (O \cup E)$ and zero, otherwise;

f_{ik} is a non-negative real variable that indicates the start time of service on node $i \in N$ by ship k .

y_{ik} is a non-negative real variable that represents the load amount in ship k at the moment immediately after the visit at node $i \in N$. In this model, for simplicity, it is assumed that the ship begins and ends empty within the planning horizon, but the model can be adapted to consider more general situations.

Mathematical model:

Minimize:

$$\sum_{i \in (S \cup O)} \sum_{j \in (O \cup E)} \sum_{k \in K} C_{ijk}^S x_{ijk} + \sum_{i \in (S \cup O)} \sum_{j \in O, D_{ij}^{dist} > 0} \sum_{k \in K} C_j^D x_{ijk} + \beta \sum_{i \in O^P} \sum_{j \in O^P, j \neq i, D_{ij}^{dist} > 0} \sum_{k \in K} x_{ijk} \quad (1)$$

subject to:

$$\sum_{j \in (O \cup E)} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in (O \cup S) \quad (2)$$

$$\sum_{i \in (O \cup S)} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in (O \cup E) \quad (3)$$

$$\sum_{j \in (O^P \cup e_k)} x_{s_k j k} = 1 \quad \forall k \in K \quad (4)$$

$$\sum_{i \in (s_k \cup O^D)} x_{i e_k k} = 1 \quad \forall k \in K \quad (5)$$

$$\sum_{i \in N} \sum_{k \in K} x_{ijk} = 0 \quad \forall j \in S \quad (6)$$

$$\sum_{j \in N} \sum_{k \in K} x_{ijk} = 0 \quad \forall i \in E \quad (7)$$

$$\sum_{i \in (O \cup S)} x_{ihk} - \sum_{j \in (O \cup E)} x_{hjk} = 0 \quad \forall h \in O; \forall k \in K \quad (8)$$

$$a_i \left(\sum_{j \in N} x_{jik} \right) \leq f_{ik} \leq b_i \left(\sum_{j \in N} x_{jik} \right) \quad \forall i \in (O \cup E); \forall k \in K \quad (9)$$

$$a_i \left(\sum_{j \in N} x_{s_k j k} \right) \leq f_{s_k k} \leq b_i \left(\sum_{j \in N} x_{s_k j k} \right) \quad \forall s_k \in (S); \forall k \in K \quad (10)$$

$$f_{jk} \geq f_{ik} + t_{ik|D_{ij}^{dist} > 0}^s + \frac{D_{ij}^{dist}}{v_k} + (x_{ijk} - 1) M \quad \forall i \in (S \cup O); \forall j \in (E \cup O); \forall k \in K \quad (11)$$

$$f_{(j+n)k} \geq f_{jk} \quad \forall j \in O^P; \forall k \in K \quad (12)$$

$$y_{jk} \geq y_{ik} + d_j + (x_{ijk} - 1) M \quad \forall i \in (S \cup O); \forall j \in (E \cup O); \forall k \in K \quad (13)$$

$$y_{jk} \leq y_{ik} + d_j + (1 - x_{ijk}) M \quad \forall i \in (S \cup O); \forall j \in (E \cup O); \forall k \in K \quad (14)$$

$$\sum_{i \in (SUO)} x_{ihk} = \sum_{j \in O} x_{j(h+n)k} \quad \forall h \in O^P; \forall k \in K \quad (15)$$

$$y_{jk} \leq C_k^{ap} \sum_{i \in (SUO)} x_{ijk} \quad \forall k \in K; \forall j \in (O \cup E) \quad (16)$$

$$\sum_{k \in K} y_{s_k k} + \sum_{k \in K} y_{e_k k} = 0 \quad \forall k \in K \quad (17)$$

$$x_{ijk} = 0 \quad \forall i, k \in A(i, k) \quad (18)$$

$$y_{jk} \leq (\alpha_{jk} C_k^{ap} + d_j) + \left(1 - \sum_{i \in (OUS_k)} x_{ijk} \right) M \quad \forall (j, k) \in O_{jk}^{FLEX}; \forall (j, k) \in A(j, k); \forall k \in K \quad (19)$$

$$y_{jk} \leq \alpha_{jk} C_k^{ap} + d_j + (1 - \alpha_{jk}) C_k^{ap} \left(1 - \sum_{i \in (OUS_k), D_{ij}^{ist} > 0} x_{ijk} \right) \quad \forall k \in K^{DP}; \forall j \in O^P \quad (20)$$

$$y_{jk} \leq \alpha_{jk} C_k^{ap} + d_j + (1 - \alpha_{jk}) C_k^{ap} \left(1 - \sum_{i \in (OUS_k), D_{ij}^{ist} > 0} x_{ijk} \right) \quad \forall k \notin K^{DP}; \forall j \notin O^{DP} \quad (21)$$

$$\sum_{i \in (O^{DP} \cup S_k)} x_{ijk} = 0 \quad \forall k \notin K^{DP}; \forall j \in O^{DP} \quad (22)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in (S \cup C), j \in (C \cup E); \forall k \in K \quad (23)$$

$$f_{ik} \geq 0 \quad \forall i \in N; \forall k \in K \quad (24)$$

$$y_{ik} \geq 0 \quad \forall i \in N; k \in K \quad (25)$$

The first part of the Objective Function (1) incorporates variable fuel costs incurred during the movement of the ships; the second part of the function represents the fixed costs of platforms berthing, and the latest part penalizes consecutive berthings at different platforms due to its high associated cost. Constraints (2) and (3) ensure that all operating sites are visited exactly once. Constraint (4) ensure that all ships leave their artificial origin depot, s_k . If a ship does not perform any visit, it leaves the depot and goes directly to its artificial final depot, e_k . Constraint (5) ensure that all ships arrive at their artificial final depot, e_k . Constraints (6) and (7) guarantee that no ship enters the artificial initial depot and no ship goes out of the artificial final depot, respectively.

Constraint (8) ensure that if a ship k has reached a node referring to an operating site, this ship must leave this node; also known as the flow conservation constraint. Constraint (9) certify that the upper and lower bounds of the time windows for both the artificial depots and the final operating sites are met. Similarly, Constraint (10) ensure that the upper and lower bounds of the time windows of origin artificial depots are met. Constraint (11) limit the exact instant in which ship k , coming from the operating site i , starts its service at the operating site j (the notation $t_{ik|D_{ij}^{ist} > 0}^s$ means that parameter t_{ik}^s is added to the right-hand side of Constraint (11) only if $D_{ij}^{ist} > 0$). These constraints avoid a sub-tour in the model's solution. Constraint (12) guarantee that a certain load will have its pickup performed before the respective delivery. Constraints (13) and (14) refer to the load balance of ship k . Note that these constraints are related to the linearization of the following equality: $x_{ijk} (y_{ik} + d_j - y_{jk})$, $\forall i \in (S \cup O), j \in (O \cup E), \forall k \in K$.

Constraint (15) ensure that if ship k picked up a load on node h (platform), then it needs to perform a visit on node $h + n$ (terminal). Constraint (16) ensure that the capacity of the ship is respected. The previously stated observation that the ship begins and ends empty is guaranteed by Constraint (17). Constraint (18) prevents ships from berthing at nodes where there are physical impediments, whether in draft, LOA or others. This constraint is guaranteed by the previous definition

of matrix $A(i, k)$. Constraint (19) limit the load in case the terminal presents draft constraints for berthing ships. Constraint (20) are used for the first case of dynamic positioning, in which there is a ship with dynamic positioning berthing in platforms. Additionally, Constraint (21) refer to the second case, in which a conventional ship berths at a platform that allows this type of ship. For the third case, in which a conventional ship cannot berth in a platform that only accepts ships with dynamic positioning, Constraint (22) are used. Finally, Constraints (23)–(25) refer to the domains of the model variables. Models (1)–(25) properly capture a number of important features of the company's practical operations. Some of them are: (i) the criteria of costs and penalties to be optimized in the ship routing; (ii) considerations of fuel consumption in terms of freighting contracts; (iii) the treatment of the platforms' inventories (tops) through the use of time windows; (iv) the high heterogeneity of the fleet and their respective capabilities and limitations; (v) rules of ship berthing at different platforms and terminals considering flexible drafts and dynamic positioning, among others.

However, there are also some simplifying assumptions that were considered in this mathematical formulation. For instance, the model does not take into account the compartmentalization of ships in order to simultaneously transport different products, but considers the ability of the ship in aggregate form. In some cases, products of different platforms may be mixed in the same compartment of the ship; some ideas for the proper extension of the model to incorporate such constraints of the incompatible mixture of products transported in a ship can be seen in [60]. The model also admits that the time windows of both the platforms and terminals are rigid because they implicitly consider the inventory constraints of ballast and the tops of the platforms, the demands of terminals, and hence, penalties for time window violations are not applied. As a result of the origin-destination approach, split loads are not allowed, a characteristic that would alter the general description and approach of the problem to a supply-demand fashion, in which the amount to be picked up or delivered at each visit in operating sites would be modeled as decision variables.

5. MIP Heuristics

5.1. Relax-And-Fix Heuristic

Heuristics based on mathematical programming, also known as MIP heuristics or primal heuristics, are procedures that can be implemented directly in the solution methods used by optimization solvers available in commercial software. The main purpose is aimed at creating alternative paths for finding relatively good integer feasible solutions in situations where exact methods are not suitable or take too much time to find an optimal solution. A widely-known MIP heuristics is the relax-and-fix heuristic proposed by [29]. According to [61], the relax-and-fix heuristic assumes that a set of binary variables y of an MIP can be partitioned into R disjoint sets of decreasing importance, given by Q^1, \dots, Q^R . Then, the corresponding R MIPs can be solved, denoted by MIP^r , with $1 \leq r \leq R$, in order to find a heuristic solution to the original MIP. On the first round, MIP^1 , an integrality condition is imposed only on variables in Q^1 , and the constraints of integrality of the remaining variables in Q^2, \dots, Q^R are relaxed. In the following MIP^r , the variables y in Q^{r-1} are fixed at their optimal values obtained in MIP^{r-1} , and the integrality constraint is added only to variables in Q^r .

MIP heuristics are widely applied in the contexts of production planning and scheduling, in which the classical partition of variables is the time index (discretized). However, it is important to emphasize that when reviewing the literature on the applications of these heuristics in the context of the routing and scheduling of vehicles, only the study presented by [52] was found. In that paper, the time was discretized to facilitate the application of the relax-and-fix heuristics, similarly to the procedures applied in problems of production planning and scheduling. The authors also claimed that they did not find any other works in the literature exploring the application of these methods in routing problems, which also motivated the present study with the application of the relax-and-fix heuristic to the continuous time problem addressed here.

The strategy adopted to apply the relax-and-fix heuristic is based on the forward time-based division of the problem, exploited by other implementations of this algorithm in other problems described in the literature. As Models (1)–(25) deal with continuous time, the time-based division can be done by arranging the time windows $[a_i, b_i]$ along the planning horizon. Figure 4 shows an illustration of the operation of the time-based forward relax-and-fix heuristic with time window partitioning. Notice that the time intervals 1, 2, . . . , n , are of different sizes, depending on the time windows chosen. Initially, pickup requests are sorted in increasing order, according to the opening of the pickup time window (a_i). Then, the m first pickup requests are selected. As these pickups are associated with their respective deliveries, the algorithm enforces the integrality of the variables of the arcs adjacent to both of these pickups and deliveries and relaxes the integrality of the variables of the arcs adjacent to the remaining pickups and deliveries, which have not yet been selected and are in the ordered list. Then, the resulting MIP, smaller than the original problem, is solved, and the variables of the arcs that are adjacent to previously-selected nodes are fixed to their integer values, except for those variables related to the arcs that connect the initial and final depots. If there are no variables with integer values, the decomposed problem is infeasible, and the algorithm stops.

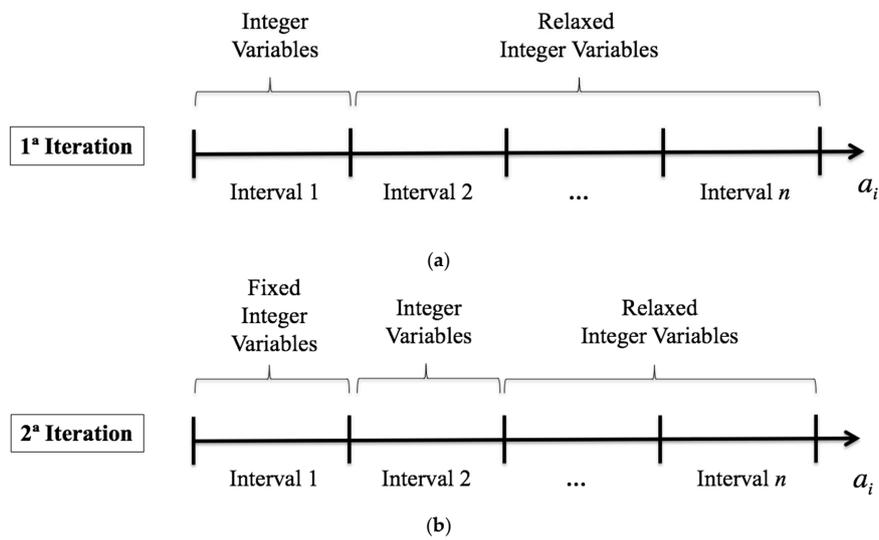


Figure 4. Schematic illustration of the operation of the relax-and-fix heuristic. (a) First iteration of the relax-and-fix procedure; (b) Second iteration of the relax-and-fix procedure.

This procedure is repeated until there are no more pickup requests in the ordered list to be allocated to ships. This procedure does not fix the variables that connect the initial and final depots of the ship along the algorithm iterations, so that the heuristic is able to insert nodes at any position of the route until the last iteration, in which the variables of the arcs of the initial and final depots are finally set to integer values. A simplified step-by-step description of the algorithm is given below.

- Step 1. Sort all pickups in ascending order of a_i ;
- Step 2. Select the m first pickups with the earliest starts a_i , which have not been selected before;
- Step 3. Enforce the integrality of the x_{ijk} variables related to the m pickups selected in Step 2 and their corresponding m deliveries, including the variables x_{ijk} with $i \in S$ or $j \in E$ related to these pickups and deliveries;
- Step 4. Relax the integrality of the x_{ijk} variables related to the pickups still not selected in Step 2, as well as to their corresponding deliveries;
- Step 5. Solve the resulting MIP;

Step 6. Fix the x_{ijk} variables resulting from the MIP solution of Step 5 to their integer values, except the variables x_{ijk} with $i \in S$ or $j \in E$ related to the m pickups selected in Step 2 and to their corresponding m deliveries;

Step 7. Repeat Steps 2–6 until there are no remaining pickups in the ordered list.

As is suggested in this paper, the relax-and-fix heuristic resembles the rolling horizon approach often used in practice (see, for instance, [62]). Other variations of the relax-and-fix heuristic could also be applied to solve this problem. For example, a variation based on the partition of the group of ships was also explored in [60], but the results were not better than the forward time-based partition based on the algorithm above.

5.2. MIP-Heuristic Based on Time Decomposition

Another simple MIP-heuristic was also developed using the same strategy of decomposing the problem into smaller sub-problems that could be solved faster. In this decomposition approach, each sub-problem considers all sub-problems of previous iterations and the variables of these (formerly solved) sub-problems that had already been fixed, just as in the relax-and-fix heuristic previously described. However, each sub-problem is now solved independently from the sub-problems of next iterations, whereas in the relax-and-fix heuristic, the future stages of the problem are taken into account in a relaxed form when solving the current stage. The objective of this decomposition strategy is to find an integer feasible solution for difficult larger problem instances; this solution may be improved using it as an initial feasible solution in some other method.

In order to decompose the original problem into several sub-problems, the developed strategy was also based on time decomposition. The sub-problems are chosen from an organized list according to the start and end of the pickup time windows in the platforms, with all ships available in each iteration. As each sub-problem takes into account all sub-problems of previous iterations, it also considers the routes previously allocated by the procedure to these sub-problems. The pseudocode of the decomposition approach is presented below.

Step 1. Set the amount of sub-problems W and calculate the amount of pickups in each iteration $w = 1, 2, \dots, W$ using $m = |O^P|/W$;

Step 2. Sort all pickups in ascending order of a_i . Break ties by choosing requests with a smaller value of $(b_i - a_i)$ first. Set $w = 1$;

Step 3. Select the first m pickups (and their respective m deliveries) that were not chosen in previous iterations from the sorted list. For all ships k , add nodes s_k and e_k ;

Step 4. Build the MIP model corresponding to the sub-problem of iteration w , composed of all pickup-delivery pairs of previously-solved sub-problems (with their routes already fixed according to Step 6), plus the m pickup-delivery pairs selected in Step 3;

Step 5. Solve this MIP model;

Step 6. Fix all routes corresponding to the solution found in Step 5, except those arcs with variables x_{ijk} with $i \in S$ or $j \in E$;

Step 7. Increment $w \leftarrow w + 1$. Repeat Steps 3–6 until the last iteration W is executed, when the variables x_{ijk} with $i \in S$ or $j \in E$ are finally fixed.

Table 1 shows an example of the decomposition of a system with nine pickups and three platforms. All of the pickups are sorted in ascending order by the opening of time windows (a_i). Notice that pickups numbered 7 and 1 have the same time window opening, so the tie is broken by choosing the earlier time window end (b_i); Pickup 7 has an earlier end is therefore chosen, after which the sequence of the decomposition is defined. In this case, the nine pickups were divided into three sub-problems with three pickups each. In the first iteration of the algorithm, the routing problem is composed of all ships and only the pickups and their respective deliveries of Sub-problem 1. In the second iteration, the MIP uses the solution of Sub-problem 1 as a fixed solution and defines the pickups and deliveries

of Sub-problem 2 as variables. The algorithm repeats this process until the last sub-problem is reached. If the sub-problem is infeasible, the algorithm stops.

Figure 5 presents the time windows of all pickups represented in Table 1. Notice that the colors show the pickups of each sub-problem used for each iteration of the algorithm.

Table 1. Example of a time decomposition.

# Pickup	Opening of Time Windows (a_i)	End of Time Window (b_i)	Platform	Sub-Problem
2	0	3	P-01	1
5	1	4	P-02	1
7	4	6	P-03	1
1	4	8	P-02	2
3	6	10	P-01	2
4	8	10	P-03	2
6	10	13	P-02	3
8	11	14	P-03	3
9	12	15	P-01	3

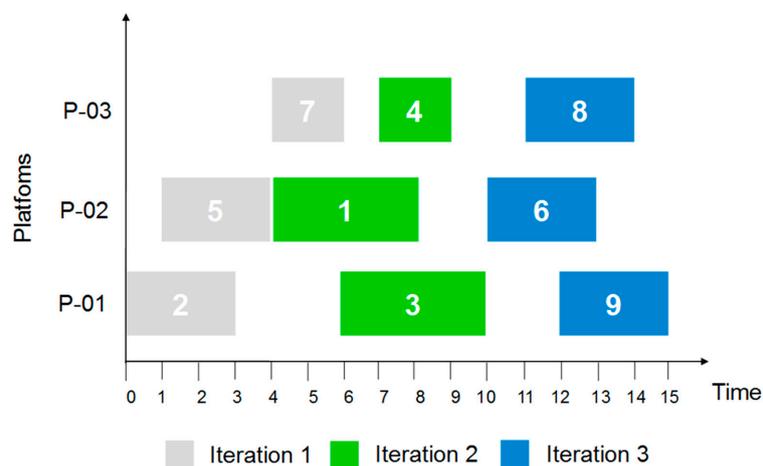


Figure 5. Decomposition of the pickups in the sub-problems.

6. Computational Results

The Models (1)–(25) of Section 3, along with the relax-and-fix and decomposition heuristics of Section 4, were initially tested by solving various problem instances of small sizes. The details of these experiments can be found in [60]. After verifying the consistency and correctness of both the model and the two heuristics in these small examples, they were tested to solve larger problem instances with real data provided by the company, with 16, 22 and 44 pairs of pickup and delivery requests, named, respectively, N16, N22 and N44, all with $|K| = 25$ ships. These instances correspond to few days of operations, depending on the instance: around three days for N16, seven days for N22 and fourteen days for N44. Table 2 presents the results of the experiments with the model.

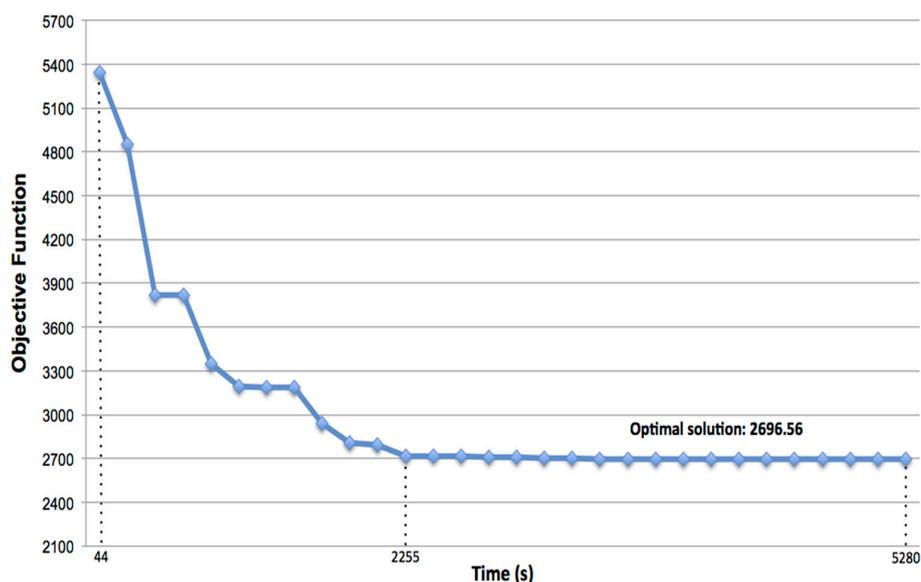
For all tests, a Dell Precision T7600 CPU E5-2680 2.70-GHz and 192 GB of RAM workstation with Windows 7 Professional operating system was used. The codes were implemented in GAMS 24.0 and solved by CPLEX 12.5.0 under academic license and using the default parameter values. The maximum time limit was set to 18,000 s (5 h), with the 32-core processing option enabled. Furthermore, other experiments were conducted exploring variations in the CPLEX parameters beyond the default values, such as: (i) the activation of local branching (LB) and relaxation-induced neighborhood search (RINS) heuristics; (ii) changes in the solver emphasis from feasibility to optimality; and (iii) shutdown of the CPLEX preprocessing procedure. However, the results obtained from these particular experiments did not significantly improve the results of Table 2.

Table 2. Results of the model using GAMS/CPLEX for problem instances N16, N22 and N44.

# Instance	# Ships Used	# Constraints	#Variables		Gap (%)	CPU Time (s)
			Integers	Continuous		
N16	10	420,930	7793	340,304	0	57
N22	13	589,362	14,482	446,504	0	5280
N44	-	1,454,611	72,543	959,104	-	-

For the instance N16, with thousands of variables and constraints, CPLEX was able to obtain an optimal solution meeting all 16 pickup and delivery requests in less than 1 min of runtime and using 10 out of the 25 available ships. For the instance N22, CPLEX was also able to obtain an optimal solution meeting all 22 pickup-delivery requests, but in a considerably longer time, 5280 s (or 1 h and 28 min) and using 13 ships out of the 25. However, increasing the number of pickup-delivery pairs to N44 (with millions of variables and constraints), CPLEX was not able to find a feasible solution within the time limit of 5 h of processing. Notice that these larger instances are extremely difficult in terms of finding a feasible solution in practice, due to the constraints regarding tighter time windows, berthing incompatibilities of several ships at different operating sites, among others. In fact, it is difficult for the company to find feasible solutions to these instances in real decision-making situations without having to partially relax some of the problem constraints.

Figure 6 depicts the variation of the model solution values over time, taking N22 as an example. Note that the first feasible solution for the model was found within 44 s. An optimal solution was obtained only after 2255 s of processing, and it was proven optimal only after 3025 s, totaling the 5280 s of processing in Table 2. This result indicates that CPLEX can find optimal solutions long before it can actually demonstrate and prove optimality.

**Figure 6.** Solution value curve of problem instance N22.

For illustration purposes, Figure 7 depicts the optimal routes found by CPLEX for the N22 example using 13 ships. Notice that the route nodes correspond to pickups in platforms (left-hand side nodes) and deliveries in terminals (right-hand side nodes), totaling 22 pairs of pickups and deliveries. Hence, consecutive pickups (deliveries) in the same platform (terminal) in Figure 7 represent in fact a single visit to the platform or terminal, so that docking costs are incurred only once. The number in parenthesis that follows the ship number is the ship capacity (in m^3), whereas the number at each node is the oil quantity loaded/unloaded in each platform/terminal. For instance, when collecting 70,000 m^3

of oil at platform 38, Ship 11 cannot be loaded with more than half of its capacity because of the dynamic positioning Constraints (20)–(22). The solutions obtained with the model were submitted to the evaluation of company experts, who indicated that the model actually represents realistic decisions within the company’s operations, and they also emphasized the high potential for practice application.

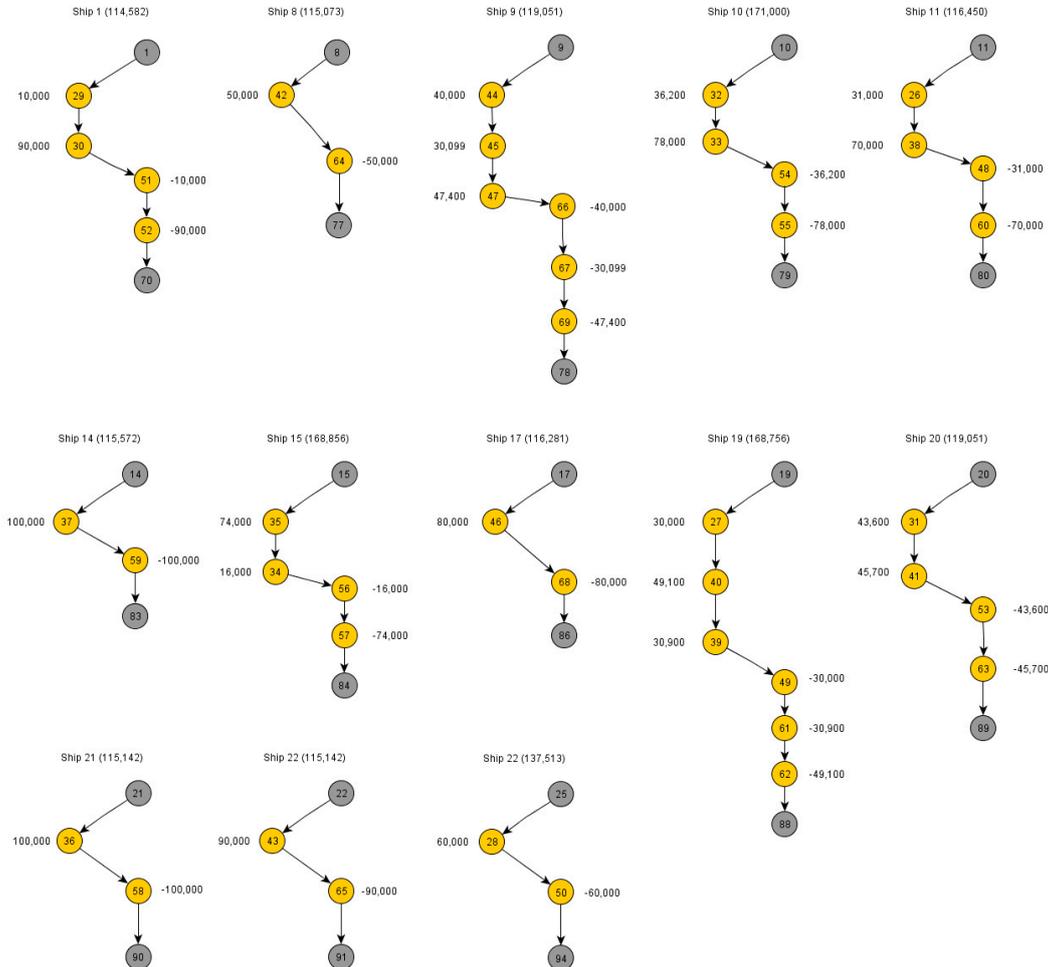


Figure 7. Optimal routes for an example in instance N22.

6.1. Computational Results from the Relax-and-Fix Heuristic

For tests with the relax-and-fix heuristic, the same problem instances N16, N22 and N44 were used. For each of them, step values (m) as multiples of the total requests amount (n), were defined, so that all iterations work with the same number of requests. The optimality gaps were also calculated based on tests N16 and N22 with the optimal solutions known from Table 2. Table 3 summarizes the results obtained for these instances with the relax-and-fix heuristic tests.

Table 3. Results of the relax-and-fix heuristic for problem instances N16, N22 and N44.

# Instance	# Ships Used	Gap (%)	CPU Time (s)
N16, $m = 8$	12	14.36	24
N22, $m = 2$	14	11.16	646
N22, $m = 11$	13	1.45	200
N44, $m = 2$	-	-	-
N44, $m = 4$	-	-	-
N44, $m = 11$	-	-	-
N44, $m = 22$	-	-	-

For instances N16 and N22, a time limit of 10 min (600 s) for each iteration of the relax-and-fix heuristic was established. For instance N16, feasible solutions were found in the last iteration of the algorithm in short computational times, however with optimality gaps larger than 10%. For instance N22, although the computational times were much lower than the computational time spent by CPLEX to optimally solve the N22 model, the optimality gaps were also greater than zero. A point to notice is that, in each iteration performed by the current implementation of the relax-and-fix heuristic, the GAMS package spent a certain amount of preprocessing time that is not directly dedicated to computational processing, accounting for about 2–10 s for each of the iterations, depending on the size of the problem. These preprocessing times could be reduced by developing a more sophisticated computational implementation.

These tests show that, as expected, the larger the m , the better the performance, both in terms of gaps and processing times. This may indicate that it is preferable to use larger steps (m) to minimize the time of re-processing in each iteration, as well as to prevent the problem from searching for local optimal solutions, represented by a smaller number of pickup and delivery requests. For the largest instance N44, no feasible solutions for any size of the partitioned problem were found with the forward time-based strategy. Since this is a relatively large and constrained problem from the perspectives of both time windows and berthing incompatibilities, it is very difficult to obtain a feasible solution for this instance. Time limits of 10, 30, 60 and 90 min for each iteration were also tested, but none of these tests resulted in feasible solutions.

Aiming at testing the capability of CPLEX to accelerate the convergence to the optimal solution of Models (1)–(25) when using an initial feasible solution, a new test was performed. The solutions with smaller gaps obtained for instances N16 and N22, respectively N16 with $m = 8$ (14.36% gap) and N22 with $m = 11$ (1.45% gap), were inserted in the CPLEX solver as initial feasible solutions for instances N16 and N22 and then run for optimality. For instance N16, the CPLEX solver with this initial heuristic solution took 70 s to find and prove the optimal solution, compared to 57 s (Table 1) required by the CPLEX solver without using this initial solution. For instance N22, CPLEX found the optimal solution and proved its optimality in only 22 s, compared to 5.280 s (Table 1). These results indicate that a good strategy for solving problems of moderate size could be to initially run the relax-and-fix heuristic and then insert the obtained solution as a starting feasible solution for the model solution via CPLEX.

6.2. Computational Tests from the Time Decomposition Heuristic

Table 4 presents the results for the time decomposition heuristic when solving the same instances of the relax-and-fix heuristic (N16, N22 and N44). This heuristic was finally capable of finding a feasible solution for the N44 instance, with $m = 11$, $m = 4$ and $m = 2$. For $m = 22$, no feasible solution was found. These solutions from the time decomposition heuristic for N44 were inserted as initial feasible solutions in Models (1)–(25), but CPLEX was unable to improve these solutions within 5 h. The results of Table 3 showed that the decomposition heuristic had a performance equal to or better than the relax-and-fix heuristic, considering that the main objective of both heuristics was to find relatively good feasible solutions quickly.

Table 4. Computational results for the time decomposition heuristic.

# Instances	Number of Iterations	# Ships Used	Gap (%)	CPU Time (s)	Objective Function
N16, $m = 8$	2	10	0.51	23	2169.76
N22, $m = 11$	2	14	13.42	45	3047.00
N22, $m = 2$	11	14	10.20	99	2960.70
N44, $m = 22$	2	-	-	-	-
N44, $m = 11$	4	21	-	1980	6489.27
N44, $m = 4$	11	22	-	247	7722.78
N44, $m = 2$	22	20	-	484	7476.33

Notice that for experiments with smaller sub-problems (smaller m), the CPU time is significantly shorter for bigger instances, like N44. The results suggest that it is faster to solve bigger instances with many smaller sub-problems than few bigger sub-problems. However, the objective function has the opposite behavior for different values of m regarding the N22 and N44 instances. Therefore, the results suggest that it is necessary to perform experimentation with each instance to find good values for m , according to the objective value and CPU time.

7. Concluding Remarks

This paper presents an optimization approach to the routing and scheduling problem of ships for oil cabotage, motivated by the practical problem of an oil company and based on an MIP model, accompanied by relax-and-fix and time decomposition heuristics. The problem is a pickup and delivery with time windows, a heterogeneous fleet, multiple depots (initial and final locations of each ship) and multiple visits at each platform and terminal over the planning horizon. Additional constraints based on business rules include operation specificities of the case under examination, as constraints related to ships with flexible drafts and terminals and devices of the dynamic positioning of ships and platforms. The MIP model captures and appropriately represents important and fundamental features of the problem within the context of oil cabotage. The model makes use of: (i) predefined time windows to relieve the oil inventory at the platforms and to meet the oil demands from the terminals; (ii) rules of berthing at different types of platforms and terminals; and (iii) considerations of fuel consumption.

The MIP model was able to obtain optimal solutions for real problems of small to moderate sizes using the GAMS/CPLEX optimization package, which is an interesting result of this study and configures a contribution for the application of such an approach within similar contexts of maritime transportation. However, the model failed to obtain feasible solutions to larger problem instances. The relax-and-fix and decomposition heuristics brought advantages in computational processing in terms of solving the model along with CPLEX and using their feasible solutions as initial solutions to the CPLEX model, although the gaps obtained with these heuristics may be larger than the expected gaps from using other heuristics, such as tabu search or ALNS, for similar maritime PDPTW. The results suggest that the proposed approach has the potential to produce reasonably good solutions to problems of moderate size in acceptable computational times in practical settings. For larger instances, only the time decomposition heuristic was capable of finding feasible solutions.

Some perspectives for future research are: (i) improvements in the relax-and-fix heuristic, such as: implementation of backward and overlapping strategies in the present time-based partition strategy; implementation of other strategies, such as partitioning ships, based on different criteria for the ordering of the list of available ships according to their costs and capabilities; (ii) improvements in the decomposition heuristic, such as: implementations exploring backward scheduling and geographical decomposition; (iii) the combination of these relax-and-fix and decomposition heuristics with local search heuristics to improve the integer feasible solutions, such as fix-and-optimize heuristics, in order to solve larger problem sets; (iv) the exploration of an alternative formulation for the pickup and delivery problem using discrete time, as opposed to continuous time, and the adaptation of the present heuristics to this alternative formulation. Additionally, further tests on the heuristics can be carried out through the use of the existing benchmark instances for maritime PDP.

Another possible future research direction would be to work on alternative modelling approaches to the problem; for example, through investigating alternative descriptions of the oil supply-demand setting, in which the oil inventory control across platforms and terminals is explicitly considered in the model and the oil quantities to be transported become decision variables, characterizing the problem as an oil inventory-routing problem [42,43,63]. For instance, see also the conversion between inventory routing problems and cargo routing problems [64]. Furthermore, the integration of this inventory-routing problem with the refineries' production planning could be explored, a feature that would provide the model with a more general characterization in terms of supply chain planning and scheduling across different decision-making hierarchy levels.

Acknowledgments: The authors thank the two anonymous reviewers for the valuable comments and suggestions for revision and the financial support of the Brazilian National Petroleum Agency (ANP), São Paulo Research Foundation (FAPESP) and the Brazilian National Council for Scientific and Technological Development (CNPq) to carry out this research.

Author Contributions: Vinicius P. Rodrigues articulated the mathematical model, designed the first versions of the heuristics and their implementation and took the lead in writing the paper. Reinaldo Morabito proposed the problem, designed the research strategy and laid out the first versions of the mathematical model. Denise Yamashita and Bruno J. V. da Silva worked on preparing and cleaning the datasets, as well as implementing and improving the models and heuristics. Paulo C. Ribas also proposed the problem and collected data for the models and heuristics, besides searching for more detailed information about the operations of oil tankers. All authors worked together in writing, revising and improving the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ANP—Brazilian National Petroleum Agency Produção Nacional de Petróleo e LGN (Barris): Electronic Spreadsheet. Available online: <http://www.anp.gov.br/?dw=1041> (accessed on 20 June 2016).
2. IEA—International Energy Agency Oil Market Report May 2016. Available online: <https://www.iea.org/media/omrreports/fullissues/2016--05--12.pdf> (accessed on 20 June 2016).
3. Petrobras Perfil—Uma Empresa Integrada de Energia. Available online: <http://www.petrobras.com.br/pt/quem-somos/perfil> (accessed on 20 June 2016).
4. Petrobras Quem Somos—Nossa História. Available online: <http://www.petrobras.com.br/pt/quem-somos/nossa-historia> (accessed on 20 June 2016).
5. Petrobras Pré-Sal: Novo Patamar de Reservas e Produção de Petróleo. Available online: <http://www.petrobras.com.br/pt/energia-e-tecnologia/fontes-de-energia/petroleo/presal> (accessed on 20 June 2016).
6. Christiansen, M.; Fagerholt, K.; Nygreen, B.; Ronen, D. Maritime Transportation. In *Handbooks in Operations Research and Management Science*; Barnhart, C., Laporte, G., Eds.; Elsevier: Amsterdam, The Netherlands, 2007; Volume 14, pp. 189–284.
7. Hennig, F.; Nygreen, B.; Christiansen, M.; Fagerholt, K.; Furman, K.C.; Song, J.; Kocis, G.R.; Warrick, P.H. Maritime crude oil transportation—A split pickup and split delivery problem. *Eur. J. Oper. Res.* **2012**, *218*, 764–774. [[CrossRef](#)]
8. Lawrence, S.A. *International Sea Transport: The Years Ahead*; Lexington Books: Lexington, MA, USA, 1972.
9. Lee, J.; Kim, B.I. Industrial ship routing problem with split delivery and two types of vessels. *Expert Syst. Appl.* **2015**, *42*, 9012–9023. [[CrossRef](#)]
10. Norstad, I.; Fagerholt, K.; Laporte, G. Tramp ship routing and scheduling with speed optimization. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 853–865. [[CrossRef](#)]
11. Christiansen, M.; Fagerholt, K.; Nygreen, B.; Ronen, D. Ship routing and scheduling in the new millennium. *Eur. J. Oper. Res.* **2013**, *228*, 467–483. [[CrossRef](#)]
12. Wang, S.; Alharbi, A.; Davy, P. Liner ship route schedule design with port time windows. *Transp. Res. Part C Emerg. Technol.* **2014**, *41*, 1–17. [[CrossRef](#)]
13. Branchini, R.M.; Armentano, V.A.; Morabito, R. Routing and fleet deployment in liner shipping with spot voyages. *Transp. Res. Part C Emerg. Technol.* **2015**, *57*, 188–205. [[CrossRef](#)]
14. Savelsbergh, M.W.P.; Sol, M. The general pickup and delivery problem. *Transp. Sci.* **1995**, *29*, 17–29. [[CrossRef](#)]
15. Furtado, M.G.S.; Munari, P.; Morabito, R. *The Pickup and Delivery Problem with Time Windows in the Oil Industry: Model and Branch-and-Cut Methods*; Federal University of São Carlos: São Carlos, Brazil, 2016.
16. Dumas, Y.; Desrosiers, J.; Soumis, F. The pickup and delivery problem with time windows. *Eur. J. Oper. Res.* **1991**, *54*, 7–22. [[CrossRef](#)]
17. Desrosiers, J.; Dumas, Y.; Solomon, M.M.; Soumis, F. Time Constrained Routing and Scheduling. In *Handbooks in Operations Research and Management Science*; Elsevier: Amsterdam, The Netherlands, 1995; Volume 8, pp. 35–139.
18. Berbeglia, G.; Cordeau, J.F.; Gribkovskaia, I.; Laporte, G. Static pickup and delivery problems: A classification scheme and survey. *Top* **2007**, *15*, 45–47. [[CrossRef](#)]
19. Nowak, M.; Ergun, O.; White, C.C. An empirical study on the benefit of split loads with the pickup and delivery problem. *Eur. J. Oper. Res.* **2009**, *198*, 734–740. [[CrossRef](#)]

20. Nowak, M.; Ergun, O.; White, C.C. Pickup and Delivery with Split Loads. *Transp. Sci.* **2008**, *42*, 32–43. [[CrossRef](#)]
21. Ropke, S.; Cordeau, J.-F. Branch-and-Cut-and-Price for the Pickup and Delivery Problem with Time Windows. *Transp. Sci.* **2009**, *43*, 267–286. [[CrossRef](#)]
22. Ropke, S.; Cordeau, J.-F.; Laporte, G. Models and a Branch-and-Cut Algorithm for Pickup and Delivery Problems with Time Windows. *Networks* **2007**, *49*, 258–272. [[CrossRef](#)]
23. Parragh, S.N.; Doerner, K.F.; Hartl, R.F. A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *J. Betr.* **2008**, *58*, 81–117. [[CrossRef](#)]
24. Bent, R.; Hentenryck, P.V. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Comput. Oper. Res.* **2006**, *33*, 875–893. [[CrossRef](#)]
25. Mitrović-Minić, S.; Krishnamurti, R.; Laporte, G. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transp. Res. Part B Methodol.* **2004**, *38*, 669–685. [[CrossRef](#)]
26. Ropke, S.; Pisinger, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transp. Sci.* **2006**, *40*, 455–472. [[CrossRef](#)]
27. Nanry, W.P.; Wesley Barnes, J. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transp. Res. Part B Methodol.* **2000**, *34*, 107–121. [[CrossRef](#)]
28. Dillenberger, C.; Escudero, L.F.; Wollensak, A.; Zhang, W. On practical resource allocation for production planning and scheduling with period overlapping setups. *Eur. J. Oper. Res.* **1994**, *75*, 275–286. [[CrossRef](#)]
29. Wolsey, L.A. *Integer Programming*; John Wiley & Sons: Hoboken, NJ, USA, 1998.
30. Toso, E.A.V.; Morabito, R.; Clark, A.R. Lot sizing and sequencing optimisation at an animal-feed plant. *Comput. Ind. Eng.* **2009**, *57*, 813–821. [[CrossRef](#)]
31. De Araujo, S.A.; Arenales, M.N.; Clark, A.R. Lot sizing and furnace scheduling in small foundries. *Comput. Oper. Res.* **2008**, *35*, 916–932. [[CrossRef](#)]
32. 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. *Comput. Oper. Res.* **2008**, *35*, 3644–3656. [[CrossRef](#)]
33. Ferreira, D.; Morabito, R.; Rangel, S. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *Eur. J. Oper. Res.* **2009**, *196*, 697–706. [[CrossRef](#)]
34. Almada-lobo, B.; Klabjan, D.; Antónia Carravilla, M.; Oliveira, J.F. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *Int. J. Prod. Res.* **2007**, *45*, 4873–4894. [[CrossRef](#)]
35. De Oliveira, L.; De Souza, C.C.; Yunes, T. Improved bounds for the traveling umpire problem: A stronger formulation and a relax-and-fix heuristic. *Eur. J. Oper. Res.* **2014**, *236*, 592–600. [[CrossRef](#)]
36. Noor-E-Alam, M.; Doucette, J. Relax-and-fix decomposition technique for solving large scale grid-based location problems. *Comput. Ind. Eng.* **2012**, *63*, 1062–1073. [[CrossRef](#)]
37. Kjeldsen, N.H.; Chiarandini, M. Heuristic solutions to the long-term unit commitment problem with cogeneration plants. *Comput. Oper. Res.* **2012**, *39*, 269–282. [[CrossRef](#)]
38. Flood, M.M. Application of Transportation Theory to Scheduling a Military Tanker Fleet. *Oper. Res.* **1954**, *2*, 150–162. [[CrossRef](#)]
39. Dantzig, G.B.; Fulkerson, D.R. Minimizing the number of tankers to meet a fixed schedule. *Nav. Res. Logist. Q.* **1954**, *1*, 217–222. [[CrossRef](#)]
40. Hoff, A.; Andersson, H.; Christiansen, M.; Hasle, G.; Løkketangen, A. Industrial aspects and literature survey: Fleet composition and routing. *Comput. Oper. Res.* **2010**, *37*, 2041–2061. [[CrossRef](#)]
41. Andersson, H.; Hoff, A.; Christiansen, M.; Hasle, G.; Løkketangen, A. Industrial aspects and literature survey: Combined inventory management and routing. *Comput. Oper. Res.* **2010**, *37*, 1515–1536. [[CrossRef](#)]
42. Christiansen, M. Decomposition of a Combined Inventory and Time Constrained Ship Routing Problem. *Transp. Sci.* **1999**, *33*, 3–16. [[CrossRef](#)]
43. Al-Khayyal, F.; Hwang, S.J. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. *Eur. J. Oper. Res.* **2007**, *176*, 106–130. [[CrossRef](#)]
44. Rocha, R.; Grossmann, I.E.; Poggi de Aragão, M.V.S. Petroleum allocation at PETROBRAS: Mathematical model and a solution algorithm. *Comput. Chem. Eng.* **2009**, *33*, 2123–2133. [[CrossRef](#)]
45. Korsvik, J.E.; Fagerholt, K.; Laporte, G. A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Comput. Oper. Res.* **2011**, *38*, 474–483. [[CrossRef](#)]

46. Bakkehaug, R.; Rakke, J.G.; Fagerholt, K.; Laporte, G. An adaptive large neighborhood search heuristic for fleet deployment with voyage separation requirements. *Transp. Res. Part C* **2016**, *170*, 129–141. [[CrossRef](#)]
47. Brønmo, G.; Christiansen, M.; Fagerholt, K.; Nygreen, B. A multi-start local search heuristic for ship scheduling—A computational study. *Comput. Oper. Res.* **2007**, *34*, 900–917. [[CrossRef](#)]
48. Korsvik, J.E.; Fagerholt, K.; Laporte, G. A tabu search heuristic for ship routing and scheduling. *J. Oper. Res. Soc.* **2010**, *61*, 594–603. [[CrossRef](#)]
49. Korsvik, J.E.; Fagerholt, K. A tabu search heuristic for ship routing and scheduling with flexible cargo quantities. *J. Heuristics* **2010**, *16*, 117–137. [[CrossRef](#)]
50. Tirado, G.; Hvattum, L.M.; Fagerholt, K.; Cordeau, J.F. Heuristics for dynamic and stochastic routing in industrial shipping. *Comput. Oper. Res.* **2013**, *40*, 253–263. [[CrossRef](#)]
51. Agra, A.; Christiansen, M.; Delgado, A.; Hvattum, L.M. A maritime inventory routing problem with stochastic sailing and port times. *Comput. Oper. Res.* **2015**, *61*, 18–30. [[CrossRef](#)]
52. Uggen, K.T.; Fodstad, M.; Nørstebø, V.S. Using and extending fix-and-relax to solve maritime inventory routing problems. *Top* **2013**, *21*, 355–377. [[CrossRef](#)]
53. Cordeau, J.F.; Laporte, G.; Potvin, J.Y.; Savelsbergh, M.W.P. Transportation on Demand. In *Handbooks in Operations Research and Management Science*; Barnhart, C., Laporte, G., Eds.; Elsevier: Amsterdam, The Netherlands, 2007; Volume 14, pp. 429–466.
54. Rakke, J.; Christiansen, M.; Fagerholt, K.; Laporte, G. The Traveling salesman problem with draft limits. *Comput. Oper. Res.* **2012**, *39*, 2161–2167. [[CrossRef](#)]
55. Todosijević, R.; Mjirda, A.; Hanafi, S.; Mladenovic, N.; Gendron, B. A General Variable Neighborhood Search for the Travelling Salesman Problem with Draft Limits. In Proceedings of the 2013 International Conference on Industrial Engineering and Systems Management (IESM), Rabat, Morocco, 28–30 October 2013.
56. Battarra, M.; Pessoa, A.A.; Subramanian, A.; Uchoa, E. Exact algorithms for the traveling salesman problem with draft limits. *Eur. J. Oper. Res.* **2014**, *235*, 115–128. [[CrossRef](#)]
57. Arnesen, M.J.; Gjestvang, M.; Wang, X.; Fagerholt, K.; Thun, K.; Rakke, J.G. A traveling salesman problem with pickups and deliveries, time windows and draft limits: Case study from chemical shipping. *Comput. Oper. Res.* **2017**, *77*, 20–31. [[CrossRef](#)]
58. FKAB A Tailor Made 3200 m³ Product Oil Tanker Design. Available online: http://www.fkab.com/no_cache/nyheter/nyhetslaesare/article/a-tailor-made-3200-m-product-oil-tanker-design.html (accessed on 13 September 2016).
59. Sørensen, A.J. A survey of dynamic positioning control systems. *Annu. Rev. Control* **2011**, *35*, 123–136. [[CrossRef](#)]
60. Rodrigues, V.P. *Uma Abordagem de Otimização Para a Roteirização e Programação de Navios: Um Estudo de Caso na Indústria Petrolífera*; Federal University of São Carlos: São Paulo, Brazil, 2014. (In Portuguese)
61. Pochet, Y.; Wolsey, L.A. *Production Planning by Mixed Integer Programming*; Springer: Berlin, Germany, 2006.
62. 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. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 896–911. [[CrossRef](#)]
63. Rani, F.K.; Rusdiansyah, A.; Wiratno, S.E.; Siswanto, N. Mixed Integer Linear Programming Model for Multi-Product Inventory Ship Routing Problem Considering Product Loading Compatibility Constraint. In Proceedings of the 11th Asia Pacific Industrial Engineering and Management Systems Conference, Melaka, Malaysia, 7–11 December 2010; pp. 7–10.
64. Stålhane, M.; Andersson, H.; Christiansen, M.; Fagerholt, K. Vendor managed inventory in tramp shipping. *Omega* **2014**, *47*, 60–72. [[CrossRef](#)]

