A Relax-and-Fix Algorithm for a Maritime Inventory Routing Problem

Marcelo W. Friske(✉) and Luciana S. Buriol

Departamento de Informática - Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil
{mwfriske,buriol}@inf.ufrgs.br

Abstract. This work presents a relax-and-fix algorithm for solving a class of single product Maritime Inventory Routing Problem. The problem consists in routing and scheduling a heterogeneous fleet of vessels to supply a set of ports, keeping inventory at production and consumption ports between lower and upper limits. Two sets of constraints are proposed both for tightening the problem relaxation and for obtaining better integer solutions. Four MIP-based local searches to improve the solution provided by the relax-and-fix approach are presented. Computational experiments were carried out on instances of the MIRPLIB, showing that our approach is able to solve most instances in a reasonable amount of time, and to find new best-known solutions for two instances. A new dataset has been created by removing the clustered characteristics of ports from the original instances, and the effectiveness of our method was tested in these more general instances.

Keywords: Maritime Inventory Routing Problem, Relax-and-Fix, MIP-Based Local Search

1 Introduction

Maritime transportation is the major mode of transportation used when considering large quantities of goods, mainly bulk products. The Maritime Inventory Routing Problem (MIRP) arises when one has to manage both the scheduling of vessels and the inventories at ports. It can be considered a variant of the Inventory Routing Problem, which combines vehicle routing and inventory management. However, MIRP deals with special features of maritime transportation.

This work considers the single product MIRP model proposed by [12]. Given a finite planning horizon, a fleet of heterogeneous vessels, and a set of ports, one must decide for each vessel which ports will be visited, when they will be visited, and the amount of product that should be loaded or discharged when a vessel operates at each port. In this problem variant, ports are grouped in geographical regions, such that each region has only production (loading) or consumption (discharging) ports. Each port has fixed storage and operating capacities, while production/consumption rates may vary along the planning horizon. The inventory of ports is supplied by vessels, and by simplified spot markets when

T. Bektaş et al. (Eds.), ICCL 2017, LNCS 10572, pp. 270–284, 2017.
https://doi.org/10.1007/978-3-319-68496-3_18

necessary. Vessels can differ by capacity, speed, and cost per sailed kilometer. The problem is classified as deep-sea, the case in which vessels spend most of the time traveling than operating at ports. The objective is to maximize the revenue of delivered products at consumption ports, subtracting travelling, operating, and spot market costs, and respecting inventory bounds of vessels and ports.

There are many opportunities for optimization considering maritime transportation. The reviews [6, 5] present a good overview of works involving optimization of maritime transportation. In [12] a good review on MIRPs models and solution methods is presented, besides proposing a core model with additional features and side constraints. They also proposed a benchmark library for the problem, called MIRPLIB [1].

The work of [4] was one of the pioneers in combining inventory managing and routing of vessels. Besides presenting an arc-flow formulation for the problem of transporting ammonia. For this problem, a path-flow formulation with coupling constraints embedded in a Branch-and-Price algorithm was proposed.

Next, we present some works that deal with deterministic and single product MIRPs. [2] proposed a discrete time fixed-charge network flow model (FCNF) for a short-sea MIRP, with variable consumption and production rates at ports. New valid inequalities generalized from the lot-sizing literature were proposed. Also, branching priorities were used for improving the search on the branch-and-bound algorithm. The FCNF model was capable of providing tight bounds and obtaining optimal solutions faster than the original formulation. [8] proposed a branch-and-price guided search for solving an extended MIRP formulation. The approach has the advantage that its components are not problem-dependent. Six local search schemes were proposed for improving the solution. Experiments have shown that the method can produce high-quality solutions quickly, even being generic. [14] proposed a framework for the inventory routing problem, which can accommodate practical features. A case study on a MIRP was done considering draft limits and minimum transport cargo for each vessel. Cuts, branching strategies, and a large neighborhood search were presented for finding optimal solutions. [9] studied MIRP models with continuous and discrete time formulations, with one or parallel docks. Experiments thereof demonstrated that continuous-time formulations can be more efficient than a discrete time model. [3] worked on a MIRP for transporting feed produced at a factory to salmon farmings in the Norwegian cost. The proposed mathematical model was reformulated for improving branch-and-bound efficiency and tightening the bounds by valid inequalities. Additionally, two matheuristics based on practical aspects of the problem were proposed for obtaining feasible solutions and for improving the current solution. [11] proposed a two-stage algorithm based on Benders decomposition for solving the deep-sea MIRP proposed in [12]. An extended time-space network was used for accommodating practical assumptions on the problem. Improvements on solutions were obtained by MIP-based local searches, branching strategies, valid inequalities, and lazy constraints. The proposed approach provided tight lower bounds and high-quality solutions in a reasonable computational time. [10] presented different matheuristics and hybrid approaches for solving a long-horizon

MIRP. Several computational experiments were performed on a set of MIRPLIB instances, and results provided new best-known values for 26 instances.

We solve a MIRP making use of a Relax-and-Fix (R&F) algorithm. R&F is a matheuristic that splits the problem into intervals or subproblems, solving them sequentially. In the first iteration, only integer variables of the first interval keep the integrality constraints. The remaining variables are relaxed. The model is then solved by a MIP solver for obtaining a partial solution. After solved, all or a part of the integer variables are fixed to their current values, and the integrality constraints are reintroduced to the variables of the next interval, resulting in a new subproblem to be solved. The algorithm iterates until there is no relaxed interval left. This technique can decompose the problem in different manners. When considering a time decomposition, the R&F has similarities with the *rolling horizon heuristic*. An overview of R&F can be found in [13].

The work of [15] applies an extended R&F algorithm on a MIRP variant, known as LNG inventory routing problem. The authors consider a structure called *end-block*, that initially simplifies or ignores part of the model for reducing the number of linear variables to be solved repeatedly.

This work presents a R&F algorithm based on the work of [15] for solving the MIRP variant presented in [12]. A set of constraints is built based on assumptions of [11], while we have proposed another set of constraints based on a assumption. They are used for tightening relaxation bounds and improving the efficiency of the algorithm. Also, four MIP-based local searches are proposed either for improving feasible solutions or removing infeasibilities. Our objective is to provide a more general method for solving MIRP instances with planning horizons up to 60 days. Although not outperforming the results of [11], our method provided new best-known values for two instances. Also, we modified the original instances in order to show that the solution approach remains effective when ports are not grouped in regions.

The remainder of this work is organized as follows. Section 2 presents the MIRP formulation and the additional constraints. In Section 3 we describe the solution method used in our computational experiments, which are presented in Section 4. Finally, Section 5 presents conclusions and future works.

2 Problem Formulation

We use the arc-flow MIRP model from [11], which is presented here for the sake of completeness. Let $\mathcal{V}$ be the set of vessels, $\mathcal{J}$ the set of ports, and $\mathcal{T}$ the set of time periods, where $T = |\mathcal{T}|$. Ports are split in subsets $\mathcal{J}^{\mathrm{P}}$ for production or loading ports ($\Delta_j = 1$), and $\mathcal{J}^{\mathrm{C}}$ for consuming or discharging ports ($\Delta_j = -1$), 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 $\mathcal{R} = \mathcal{R}^{\mathrm{P}} \cup \mathcal{R}^{\mathrm{C}}$. The discrete time model is built under a port-time structure, composed of a set of nodes and a set of directed arcs. Each vessel $v \in \mathcal{V}$ has its own arc set $\mathcal{A}^v$, while the nodes set is shared by all vessels. Regular port-time nodes $n = (j, t) \in \mathcal{N}$ represent a possible operation (loading or discharging) by a vessel at port $j \in \mathcal{J}$

at time $t \in \mathcal{T}$. Node set $\mathcal{N}_{0,T+1}$ is composed by set $\mathcal{N}$, a source node n_0, and a sink node n_{T+1}, which represent the starting and ending positions of each vessel in the system, respectively. Each arc set $\mathcal{A}^v$ is composed by five arc types. A source arc $a = (n_0, (j,t))$ links the source node to the initial vessel position, arriving at port j at time period t. Traveling arcs $a = ((j_1, t_1), (j_2, t_2))$ represent a voyage that departs from port j_1 at time t_1 and arrives at port j_2 at time t_2, such that $j_1 \neq j_2$. Waiting arcs $a = ((j,t), (j, t+1))$ represent that the vessel remains at the same port j at times t and $t+1$. Sink arcs $a = ((j,t), n_{T+1})$ link a regular node to a sink node, for vessels that exit the system at port j at time t. Arc $a = (n_0, n_{T+1})$ links source and sink nodes for unused vessels. We ignore this arc as the instances proposed in [12] consider that all vessels are used.

Binary variable x_a^v is set to 1 if vessel v travels along arc $a \in \mathcal{A}^v$, and binary variable z_{jt}^v is 1 if vessel v operates (discharge or load product) at port j in time t. Continuous variables s_{jt} and s_t^v represent the inventory of port j and vessel v at the end of time period t, respectively. Variables f_{jt}^v represent the amount loaded or discharged at port j in time period t by vessel v. Variable α_{jt} is the amount of product sold to or bought from a spot market by port j at time period t. The single product MIRP can be modeled as follows:

$$\max \sum_{j \in \mathcal{J}^C} \sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}} R_{jt} f_{jt}^v - \sum_{v \in \mathcal{V}} \sum_{a \in A^v} C_a^v x_a^v - \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}} (t\epsilon_z) z_{jt}^v - \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} P_{jt} \alpha_{jt} \quad (1)$$

$$\text{s.t.} \quad \sum_{a \in FS_n^v} x_a^v - \sum_{a \in RS_n^v} x_a^v = \begin{cases} +1 & \text{if } n = n_0, \\ -1 & \text{if } n = n_{T+1}, \\ 0 & \text{if } n \in \mathcal{N}, \end{cases} \quad \forall\, n \in \mathcal{N}_{0,T+1}, v \in \mathcal{V} \quad (2)$$

$$s_{jt} = s_{j,t-1} + \Delta_j \left(d_{jt} - \sum_{v \in \mathcal{V}} f_{jt}^v - \alpha_{jt} \right), \quad \forall\, n = (j,t) \in \mathcal{N} \quad (3)$$

$$s_t^v = s_{t-1}^v + \sum_{\{n=(j,t) \in \mathcal{N}\}} \Delta_j f_{jt}^v, \quad \forall\, t \in \mathcal{T}, v \in \mathcal{V} \quad (4)$$

$$\sum_{v \in \mathcal{V}} z_{jt}^v \leq B_j, \quad \forall\, n = (j,t) \in \mathcal{N} \quad (5)$$

$$z_{jt}^v \leq \sum_{a \in RS_n^v} x_a^v, \quad \forall\, n = (j,t) \in \mathcal{N}, v \in \mathcal{V} \quad (6)$$

$$s_t^v \geq Q^v x_a^v, \quad \forall\, v \in \mathcal{V}, a \in \mathcal{A}_{\mathrm{PC}}^v, \quad (7)$$

$$s_t^v \leq Q^v (1 - x_a^v), \quad \forall\, v \in \mathcal{V}, a \in \mathcal{A}_{\mathrm{CP}}^v, \quad (8)$$

$$\sum_{t \in \mathcal{T}} \alpha_{jt} \leq \alpha_j^{\max}, \quad \forall\, j \in \mathcal{J} \quad (9)$$

$$0 \leq \alpha_{jt} \leq \alpha_{jt}^{\max}, \quad \forall\, j \in \mathcal{J}, t \in \mathcal{T} \quad (10)$$

$$F_{jt}^{\min} z_{jt}^v \leq f_{jt}^v \leq F_{jt}^{\max} z_{jt}^v, \quad \forall\, n = (j,t) \in \mathcal{N}, v \in \mathcal{V} \quad (11)$$

$$S_j^{\min} \leq s_{jt} \leq S_j^{\max}, \quad \forall\, n = (j,t) \in \mathcal{N} \quad (12)$$

$$0 \leq s_t^v \leq Q^v, \quad \forall\, v \in \mathcal{V}, t \in \mathcal{T} \quad (13)$$

$$x_a^v \in \{0,1\}, \quad \forall\, v \in \mathcal{V}, a \in A^v; \quad z_{jt}^v \in \{0,1\}, \ \forall n = (j,t) \in \mathcal{N}, v \in \mathcal{V}\ . \quad (14)$$

Objective function (1) maximizes the revenue R_{jt} of the unloaded product at discharging ports, subtracting arc costs C_a^v used by each vessel. The third term is an additional value that induces vessels to operate as soon and as few times as possible. The penalization value P_{jt} for using spot markets is accounted in the last term of the equation. Constraints (2) refer to the flow balance of vessels along the nodes, where FS^v_n and RS^v_n refer to the set of outgoing and incoming arcs associated with node $n \in \mathcal{N}_{0,T+1}$ and vessel $v \in \mathcal{V}$, respectively. Constraints (3) define ports inventory balance at the end of each time period, where d_{jt} represents the production/consumption rate of port j in time period t. Constraints (4) refer to the vessels inventory balance at the end of each time period. Constraints (5) limit to B_j (berth limit) the number of vessels that can operate simultaneously at a node. Constraints (6) require that a vessel can only operate at a node if it is actually at that node. Constraints (7) require that the vessels must travel at the maximum capacity when traveling from a loading port to a discharging port or to the sink node, where $\mathcal{A}^v_{\text{PC}} = \{a = ((j_1, t), (j_2, t')) \in \mathcal{A}^v : j_1 \in \mathcal{J}^{\text{P}}, j_2 \in \mathcal{J}^{\text{C}} \cup \{n_{T+1}\}\}$. Constraints (8) require that a vessel must be empty when traveling from a discharging port to a loading port or to a sink node, where $\mathcal{A}^v_{\text{CP}} = \{a = ((j_1, t), (j_2, t')) \in \mathcal{A}^v : j_1 \in \mathcal{J}^{\text{C}}, j_2 \in \mathcal{J}^{\text{P}} \cup \{n_{T+1}\}\}$. Constraints (10) limit to $\alpha^{\max}_{jt}$ the amount of products sold to or bought from spot markets by a port in each time period, and (9) limit to $\alpha^{\max}_j$ the cumulative amount for using spot markets. Constraints (11) impose that the amount loaded/discharged by a vessel at a port must lie between $F^{\min}_{jt}$ and $F^{\min}_{jt}$ in each time period. Constraints (12) assure that ports inventory must lie between lower $S^{\min}_j$ and upper $S^{\max}_j$ limits in each time period. Constraints (13) limit the vessel inventory to its capacity Q^v. Finally, (14) restricts the variables x^v_a and z^v_{jt} to be binaries.

2.1 Additional Constraints

In this section we consider simplifying assumptions that lead to two sets of constraints for the presented MIRP. They are useful for tightening the lower bound and for accelerating the relax-and-fix approach. We proposed the first set based on the following assumption: considering a small vessel, which capacity Q^v is less or equal to $F^{\max}_{jt}$ for some $j \in \mathcal{J}$ and $t \in \mathcal{T}$, then it can fully load or discharge in just one time period at port j. Equation (15) imposes that if a vessel operates at a port in a time period, it must leave the port in the same time period. This assumption allows a vessel to be available for more voyages, avoiding that it waits at a port after finishing its operation.

$$\sum_{a \in FS^v_{n'}} x^v_a = z^v_{jt} > \forall j \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} : Q^v < F^{\max}_{jt} \quad . \tag{15}$$

In Eq. (15), $FS^v_{n'} \subseteq FS^v_n$ is the set of outgoing arcs from node $n = (j, t)$ for vessel v which arrives at a port of different type, or arrives at the sink node.

One may ask if constraints (15) do not cut a possible optimal solution in which a small vessel may split its inventory, operating consecutively at two ports of the same region. However, fractioning a vessel inventory between two or more ports in a region means that a smaller amount will be discharged or loaded at these ports. Therefore, the stocks at ports will reach their lower/upper limits sooner, requiring that another vessel operates at these ports sooner, too. This "premature" visit incurs additional costs and may be avoided by forbidding split inventory of small vessels.

The second set of constraints is based on the "Two-port-with-no-revisit" assumption of [11]. It assumes that if a vessel arrives at 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. [11] developed an augmented time-space network that easily implements this assumption. However, implementing the assumption directly on model (1) requires additional sets of binary variables and side constraints that increase substantially the size of the model, making it more difficult to solve. We then propose the constraints below:

$$\sum_{j \in \mathcal{J}_r} \sum_{t \in \mathcal{T}} \sum_{a \in FS^v_{n\text{intra}}} x^v_a \leq \sum_{j \in \mathcal{J}^r} \sum_{t \in \mathcal{T}} \sum_{a \in RS^v_{n\text{inter}}} x^v_a, \;\; \forall v \in \mathcal{V}, r \in \mathcal{R} \; . \tag{16}$$

In Eq. (16), $\mathcal{J}^r$ is the set of ports of a region $r \in \mathcal{R}$, $FS^v_{n\text{intra}}$ is the set of intra-regional arcs of vessel v that depart from node n, and $RS^v_{n\text{inter}}$ is the set of inter-regional arcs of vessel v that arrives at node n. Constraints (16) ensure that the number of selected intra-regional arcs will be less or equal to the number of entering arcs for each region and each vessel. The constraint is partially effective when considering more than one visit to region r of vessel v. This occurs because 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 the constraints (16). This occurs because the constraints do not consider each visit of a vessel to a region but the entire time horizon.

3 The Proposed Relax-and-Fix Approach

In the R&F, the planning horizon $\mathcal{T}$ is divided in p intervals, where $\mathcal{I} = \{1, \ldots, p\}$ is the set of all intervals. Each interval $i \in \mathcal{I}$ corresponds to all variables and constraints that have a time index $t \in \{\frac{T}{p}(i-1), \ldots, \frac{T}{p}i\}$, such that $T \bmod p = 0$.

Figure 1 illustrates the first, second, and last iterations of the R&F, considering a network structure for a single vessel, divided in $p = 4$ intervals. At the first iteration ($it = 1$, Fig. 1-(a)), binary variables x^v_a and z^v_{jt} of the interval $i = it$ are restricted to be integer. This interval belongs to the "integer block". The remaining intervals have their integer variables relaxed, belonging to the "relaxed block". The last two intervals ($e = 2$) are omitted from the problem. These intervals belong to the called "end-block" [15], subject to $e \leq p - 2$. A MIP solver is then used for solving the current problem. At iteration $it = 2$ (Fig. 1-(b)), binary variables of interval $i = it - 1$ are fixed with the values obtained in the previous iteration, now belonging to the "fixed block". Original continuous variables of model (1)-(14) are kept unfixed in all iterations. Integrality constraints are reintroduced into the variables of the interval $i = it$. Also, one interval from the end-block turns to belong to the relaxed block. Then, the problem is solved again by the MIP solver. The algorithm continues iterating until $it = p$, i.e. all intervals have been removed from the end-block and integrality constraints are reintroduced to the variables of all intervals (Fig. 1-(c)). At this point, a solution for the original problem is then returned.

According to Fig. 1, arc variables x_a^v have a special treatment in the R&F when they cross two different blocks. For example, let $a = ((j_1, t_1), (j_2, t_2))$ be a travel arc crossing two different blocks. We consider that the block in which time t_2 belongs has dominance over the block in which t_1 belongs. This rule does not apply to the source arcs (that are originally fixed) and sink arcs. Sink arcs are never fixed in the R&F. This occurs because if a sink arc variable is set to 1 and fixed for some vessel, it will not be available in the remaining time horizon, which can lead to an infeasible solution. On the other hand, if sink arcs are fixed to 0, this implies that the vessel remains available in the system when maybe it is no longer necessary, impacting on the objective function value.

In the relax-and-fix strategy, solving each interval up to optimality does not necessarily lead to an optimal solution for the original problem. In this case, we use MIP relative GAP and time limit as stopping criteria in each iteration, as suggested in [15]. Initially, the MIP relative GAP is set to a positive value, which is linearly decreased along the iterations such that in the last iteration the MIP relative GAP is set to 0.0%.

During the R&F iterations, it is possible that the problem becomes infeasible when an interval is fixed and integrality constraints are reintroduced into the next interval. A common approach for avoiding infeasibility is to use an overlap which does not fix part of the integer interval at each iteration [13]. In our case the overlap just reduces the size of the fixed block, leading to more integer variables to be solved along the iterations.

Even using overlap, port-time inventory bounds can be violated. It occurs when no vessel can reach a port at specific times due to the previously fixed routing decisions and the spot market variables are not sufficient to avoid lack or surplus of inventory. To handle this issue, we introduce auxiliary variables $\beta_{jt} \geq 0, j \in \mathcal{J}, t \in \mathcal{T}$. These variables work as an unlimited spot market and are highly penalized in the objective function. Eq. (3) is reformulated:

$$s_{jt} = s_{j,t-1} + \Delta_j \left(d_{jt} - \sum_{v \in V} f_{jt}^v - \alpha_{jt} - \beta_{jt} \right), \quad \forall n = (j,t) \in \mathcal{N} \; . \tag{17}$$

Note that the use of auxiliary variables avoids the solver to stop prematurely, but if a variable $\beta_{jt}, j \in \mathcal{J}, t \in \mathcal{T}$ is positive at the end of R&F, the solution for the original problem remains infeasible.

3.1 Improvement Phase

MIP-based local searches are applied on the solution returned by the R&F algorithm for improving the solution quality, removing possible infeasibilities. MIP-based local search is an effective method which has been used in several works,

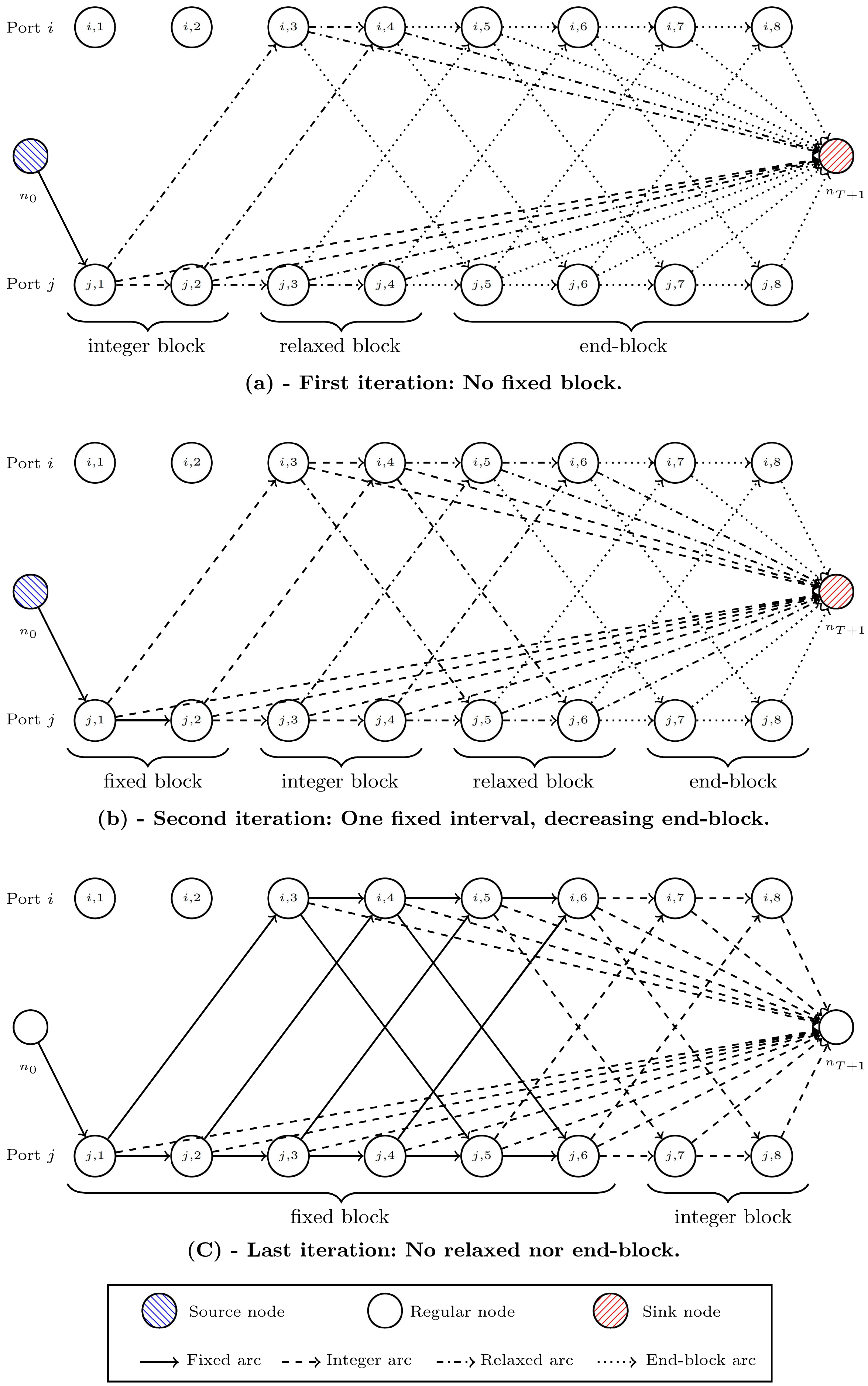

Fig. 1. First, second, and last iteration of relax-and-fix for a network of a single vessel

including MIRPs [14, 7, 8, 11, 3]. We describe four MIP-based local search procedures. They fix all integer variables from the solution obtained by R&F and iteratively allow a set of these variables to be optimized. Continuous variables are always free to be optimized in all approaches.

1. **Improving Time Intervals.** This procedure consists of dividing the time horizon into m intervals, such that $k = \{1, \ldots, m\}$ is the number of iterations, one for each interval. At each iteration, the integer variables of interval k are unfixed, following the same rules adopted for the R&F. After being optimized, theses variables are fixed to the newly obtained values. This procedure repeats iteratively until no improvement is achieved by optimizing at least one interval in the m iterations.
2. **Improving Vessels Pairs.** Following the idea of [7] which explores the neighborhood between two vessels, this procedure consists in iteratively selecting a pair of vessels to be optimized. Let v_1 and v_2 be the vessels selected to be optimized in an iteration. Then, variables x_a^v and z_{jt}^v, such that $a \in \mathcal{A}^v, j \in \mathcal{J}, t \in \mathcal{T}, v \in \mathcal{V} : v = \{v_1, v_2\}$ are unfixed. Vessels are selected at random with no repetitions. The algorithm runs until no improvement is achieved for $\binom{|\mathcal{V}|}{2}$ iterations. As the number of pairs grows considerable in large instances, after all pairs were tested once, the stopping criteria is changed to $|\mathcal{V}|$ iterations without improvement.
3. **Improving Vessels and Time Intervals.** This improvement approach can be viewed as a combination of the two previous methods. The time horizon is divided into m intervals, allowing one interval to be optimized at a time. Also, all integer variables corresponding to a vessel are allowed to be optimized per iteration. After optimizing a solution, all integer variables of this vessel are fixed to the new values, except those belonging to the interval which is being optimized. Then, the next vessel and the same time interval is optimized. The algorithm iterates between all time intervals and all vessels, $m|\mathcal{V}|$ steps in each iteration. The search stops when no improvement is achieved in one complete iteration.
4. **Improving Port Types.** This procedure is suggested by [11] as an extension of the “Fix Supply” and “Fix Demand” proposed in [8]. First, all integer variables associated with the loading ports are fixed, while integer variables associated with discharging ports are optimized. Then, variables of discharging ports are fixed to the new values, and the variables of loading ports are optimized. Variables that correspond to arcs that connect ports of different types are kept unfixed in the whole procedure for allowing a vessel to depart from a region earlier if possible. According to [11], optimizing first discharging ports is justified due to the fact that 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 MIP relative GAP as stopping criteria.

4 Computational Results

This section presents computational results obtained by solving the MIRP model with the algorithms described in Sect. 3 As in [11], we solved the model as a minimization problem, turning the objective function (1) negative. The algorithms were implemented using CPLEX 12.5 C++ API and compiled with the optimization parameter −O3. Experiments were carried out on a AMD-FX-8150 computer running at 3.6 GHz on a single core, with 32 GB RAM.

4.1 Benchmark Instances

Computational results were performed on "Group 1" instances available in the MIRPLIB [1]. The instances name present their characteristics. For example, instance "LR2_11_DR2_22_VC3_V6a" means that there exists 2 loading regions (LR), and in each region there is one loading port, two discharging regions (DR), each of them with 2 ports, three vessel classes (VC), and a total of six available vessels (V), at least one for each vessel class. The letter at the end of the name is used for differentiating instances with the same size. Each instance was tested with time horizons of 45 and 60 days, with time periods of one day.

Modified Instances For removing the clustered characteristic of ports, we modified the MIRPLIB instances concerning port coordinates and if necessary, production/consumption rates. Usually, ports are grouped in regions, especially in deep-sea configuration. However, it seems natural that there may exist cases where each region has just one port, or ports in the same region are not necessarily of the same type (loading or discharging). Let x_j and y_j be the coordinates of each port in the original instance. Also, let $\overline{x} = \max_{j \in \mathcal{J}}\{x_j\}$, $\overline{y} = \max_{j \in \mathcal{J}}\{y_j\}$, $\underline{x} = \min_{j \in \mathcal{J}}\{x_j\}$, and $\underline{y} = \min_{j \in \mathcal{J}}\{y_j\}$ be the extreme coordinates of the instance. Then, for each port $j \in \mathcal{J}$ we define the new coordinates at random as follows: $x_j = rand(\underline{x}, \overline{x})$ and $y_j = rand(\underline{y}, \overline{y})$. The seed value used for each instance was $\overline{x}$. The distances between ports and cost of arcs are recalculated according to [12]. Instance "LR2_22_DR2_22_VC3_V10a" turns infeasible due to the new ports positions. For this case, the values of $d_{jt}, j \in \mathcal{J}, t \in \mathcal{T}$ were reduced by 10%. The modified instances are available in the author's web page[1].

4.2 Parameters and Methodology

From initial experiments with parameters that seemed to be promising, we built a methodology for the computational experiments. First, 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). The difficulty of solving the instances can be evaluated by other characteristics (number of variables/constraints, average port capacity-to-rate ratio $\frac{S^{\max}_{jt}}{d_{jt}}$ [12], among others), but we prefer a first

[1] http://inf.ufrgs.br/∼mwfriske

simple classification to consider different parameters for each set. Also, we do not distinguish the difficulty considering the time horizon of the same instance, i.e. the parameter for an instance with $T = 45$ will be the same for the correspondent instance with $T = 60$. The exception occurs with the number of intervals p that the time horizon is divided in the R&F. Parameters and possible values tested for each set are described in Table 1.

Table 1. Parameter values used in the computational experiments

	Name	**Acronym**	**Value** set-1	set-2	
Relax-and-Fix	Number of intervals	p	{5,9}		Case $T = 45$
			{6,10}		Case $T = 60$
	Overlap (%)	o	{15,30,50}		
	Time limit for solving each interval (s)	$t_{\text{rf}}^{\text{it}}$	{50,100,200}	{100,200,400}	
Local search	Time limit for iteration (s)	$t_{\text{ls}}^{\text{it}}$	25	50	Not using β_{jt}
			35	75	Using β_{jt}
			70	150	
			140	300	
	Time limit for the entire local search (s)	$t_{\text{ls}}^{\max}$	7200	10800	

According to Table 1, each instance set can have more than one value for each parameter. We first test the smallest value for each parameter, and when necessary they 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 turns 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. Even with the maximum values of o and $t_{\text{rf}}^{\text{it}}$, if no integer solution was found, or solutions remained infeasible, we change the value of p from 5 to 9, and reset the other parameters to the minimum values, increasing them if necessary. If no solution has been found by varying the previous parameters, we added to the model the auxiliary variables β_{jt}, again resetting p, o and $t_{\text{rf}}^{\text{it}}$ to its minimum values. When using auxiliary variables, $t_{\text{ls}}^{\text{it}}$ is also increased. If a solution remains infeasible during R&F or at the end of the local search, o, $t_{\text{rf}}^{\text{it}}$ and $t_{\text{ls}}^{\text{it}}$ are increased together. At this point, we stopped the tests, even if no feasible solution was found.

The number of intervals in the end-block at starting the R&F algorithm is always set to $p - 2$ in order to solve a minimum number of continuous variables per iteration, saving computational time. The initial optimality GAP is set to 50%. For the local search procedures which divide the planning horizon in m intervals (see items 1 and 3 of Sect. 3.1), we set $m = 3$. For all local search procedures, the optimality GAP is set to 0.1%.

4.3 Lower Bounds

For evaluating the effectiveness of the proposed additional constraints, the lower bounds were computed solving the relaxation of model (1) with and without the additional constraints. We consider the values that were obtained after the solver performed the cuts in the first node of the branch-and-cut tree. Considering the MIRPLIB instances, lower bounds improved on average 46.5% and 6.4% for instances of sets 1 and 2, respectively. This improvement is solely due to the first set of constraints. A reason for improving the bounds is that forcing some vessels to depart from the port to another type of port after the operations avoids the use of waiting arcs, which have no cost. The major effectiveness of the constraints on instances of set-1 may occur because the vessels will be forced to depart in direction to just one region. Then, if a vessel uses fractions of arcs when traveling to another region, as these arcs have similar costs, the relaxation value can be better. On the other hand, since in set-2 one must decide between ports grouped in at least two separated regions, consequently, a fractional solution may use arcs with a large cost difference. When considering the modified instances, the constraints improved the lower bounds in 15.3% for set-1 and 11.3% for set-2. The minor improvement on lower bounds considering instances of set-1 occurs because the ports of the same type are not grouped into regions.

4.4 Relax-and-Fix and Improvement Phase Results

This section presents the results obtained using the relax-and-fix algorithm and the MIP-based local searches. Two combinations of local searches were tested. Combination A uses procedures 1, 2 and 4 from Section 3.1, respectively, while combination B uses procedures 3 and 4, respectively. We present only the results considering the combination B, as it performed better in most of the instances than the combination A. The time limit $t_{\text{ls}}^{\max}$ 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(es).

Table 2 presents the results of the MIRPLIB and modified instances. Column "Parameters" presents the parameter values, columns "R&F" present the results considering only the R&F algorithm, while columns "LS" present the results concerning the performed local search in the R&F solution. Columns "BKV" present the best-known values of MIRPLIB instances obtained by [11]. Column "Obj" corresponds to the objective value, and column "Time" corresponds to the total processing time in seconds. The processing time of [11] was normalized using the *PassMark Software* [2]. Column "GAP$^{\text{BKV}}$" presents the relative deviation $(\frac{Obj-BKV}{-BKV}) * 100$, where Obj corresponds to the objective value of our algorithm, while BKV corresponds to the objective value of [11]. Column "GAP$^{\text{LB}}$" corresponds to the relative deviation $\frac{Obj-LB}{-LB}$ from the lower bound LB. The average values do not include instances where the relative deviation is labeled as "-", meaning that no feasible solution was found.

MIRPLIB Instances Results. According to Table 2, the average time for obtaining the corresponding solutions is shorter than the time reported by [11].

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

Table 2. MIRPLIB and modified instance results

Instance	T	MIRPLIB Instances: Parameters $\beta_{j,t}$	ρ	o	t_{rf}^{it}	t_{ls}^{it}	R&F Time	R&F Obj	LS Time	LS Obj	GAP^{BKV}	BKV Time	BKV Obj	Modified instances: Parameters $\beta_{j,t}$	ρ	o	t_{rf}^{it}	t_{ls}^{it}	R&F Time	R&F Obj	LS Time	LS Obj	GAP^{LB}
LR1_1_DR1_3_VC1_V7a	45		5	15	50	25	55	-13.178	454	-13.272	**0,00%**	173	-13.272		5	15	50	25	12	-19.547	986	-21.491	5,51%
LR1_1_DR1_4_VC3_V11a	45		5	15	50	25	159	-10.682	822	-10.910	2,93%	8.362	-11.239		5	15	50	25	65	-22.164	2.556	-24.617	8,02%
LR1_1_DR1_4_VC3_V12a	45		5	15	50	25	38	-8.540	624	-10.372	3,35%	7.828	-10.732		9	30	50	25	136	-22.263	1.618	-23.062	6,07%
LR1_1_DR1_4_VC3_V12b	45		5	15	50	25	179	-7.999	1.611	-9.057	0,14%	1.742	-9.069		5	15	50	25	129	-23.883	3.625	-26.842	5,87%
LR1_1_DR1_4_VC3_V8a	45		9	30	50	25	26	-4.688	86	-5.106	**0,00%**	4.517	-5.106		5	15	50	25	79	-15.731	2.797	-17.342	13,00%
LR1_1_DR1_4_VC3_V9a	45		5	15	50	25	29	-5.419	649	-6.629	3,80%	502	-6.891		5	15	100	25	123	-16.196	2.101	-17.324	14,07%
LR1_2_DR1_3_VC2_V6a	45		5	30	50	25	256	-9.511	1.797	-10.577	5,00%	5.844	-11.134		5	15	50	25	43	-17.896	1.451	-19.597	8,79%
LR1_2_DR1_3_VC3_V8a	45		5	15	50	25	188	-10.133	1.255	-11.680	2,75%	7.482	-12.010		5	15	100	25	125	-19.138	1.827	-20.568	14,78%
LR2_11_DR2_22_VC3_V6a	45	•	5	50	400	300	1.610	94.720	5.401	-9.550	1,73%	7.987	-9.718		5	15	100	50	171	-14.017	5.410	-15.064	27,22%
LR2_11_DR2_33_VC4_V11a	45	•	5	15	100	75	395	565.310	5.475	-13.218	5,70%	8.736	-14.017		9	15	30	50	144	-24.328	5.450	-27.728	21,60%
LR2_11_DR2_33_VC5_V12a	45	•	5	15	100	75	435	747.441	5.551	-15.125	17,90%	8.756	-18.423	•	5	15	100	75	441	925.516	5.852	-22.581	46,39%
LR2_22_DR2_22_VC3_V10a	45	•	5	50	400	300	2.035	1.374.400	5.436	-21.957	11,43%	8.552	-24.789	•	5	50	400	300	2.041	1.397.820	6.002	-23.326	33,86%
LR2_22_DR3_333_VC4_V14a	45	•	5	15	100	75	585	1.825.910	6.183	25.843	-	9.219	-21.952	•	5	15	100	75	597	891.797	5.706	-28.232	41,41%
LR2_22_DR3_333_VC4_V17a	45	•	5	50	400	300	2.392	6.723.570	9.047	2.033	-	9.337	-21.713	•	5	15	100	75	756	8.208.870	5.941	62.221	-
Average							599		3.171		4.56%	6.360							347		3.666		18,97%
LR1 1 DR1 3 VC1 V7a	60		6	50	50	25	176	-16.326	776	-16.675	**0,00%**	435	-16.675		6	15	50	25	69	-24.995	1.584	-27.275	6,16%
LR1_1_DR1_4_VC3_v11a	60		6	15	50	25	214	-11.113	2.312	-11.516	13,13%	6.997	-13.257		6	15	50	25	148	-26.952	3.625	-31.455	7,92%
LR1_1_DR1_4_VC3_V12a	60	•	6	15	50	35	190	-10.012	1.584	-11.223	0,41%	7.828	-11.269	•	6	30	50	35	198	-27.495	3.564	-29.613	2,89%
LR1_1_DR1_4_VC3_V12b	60		6	15	100	25	481	-8.018	2.960	-9.958	0,94%	8.537	-10.053		6	15	50	25	165	-33.163	3.625	-34.264	6,38%
LR1_1_DR1_4_VC3_V8a	60	•	6	15	50	35	115	325.680	865	-4.578	11,81%	7.534	-5.191		6	15	100	25	214	-19.355	2.552	-20.905	17,89%
LR1_1_DR1_4_VC3_V9a	60		6	15	50	25	178	-6.746	757	-6.904	8,58%	8.120	-7.552		6	15	100	25	172	-18.922	3.701	-21.640	16,99%
LR1_2_DR1_3_VC2_V6a	60		10	15	30	25	172	-10.514	1.869	-12.639	7,28%	8.135	-13.631		6	15	50	25	165	-22.908	2.377	-25.324	9,85%
LR1_2_DR1_3_VC3_V8a	60		6	15	200	25	495	-12.857	1.573	-14.329	4,04%	7.871	-14.931		6	15	100	25	277	-23.370	1.827	-25.687	17,35%
LR2_11_DR2_22_VC3_V6a	60	•	6	15	100	75	245	195.984	5.403	39.102	-	8.237	-13.351	•	6	15	100	75	432	625.009	5.701	-19.230	27,67%
LR2_11_DR2_33_VC4_V11a	60	•	6	15	100	75	560	1.523.420	5.702	181.868	-	8.765	-17.008	•	6	15	100	75	660	504.184	5.551	-32.807	29,46%
LR2_11_DR2_33_VC5_V12a	60	•	6	15	100	75	601	906.791	5.702	176.226	-	8.824	-22.730	•	6	50	400	300	2.554	2.632.490	7.204	173.385	-
LR2_22_DR2_22_VC3_V10a	60	•	6	15	100	75	639	2.685.950	5.823	826.372	-	8.628	-32.627	•	6	15	100	75	604	1.514.310	5.787	92.707	-
LR2_22_DR3_333_VC4_V14a	60	•	6	15	100	75	1.094	7.925.830	6.222	2.218.890	-	9.352	-26.873	•	6	30	200	150	2.030	2.728.360	5.863	517.543	-
LR2_22_DR3_333_VC4_V17a	60	•	6	30	200	150	2.134	4.228.370	5.732	1.110.680	-	9.424	-27.000	•	6	30	200	150	2.526	8.152.140	6.653	3.014.670	-
Average							521		3.377		5.77%	7.763							730		4.258		14,25%

The relative gap to the BKV was on average 2.2% for set-1 and 9.2% for set-2 when $T = 45$, and considering instances in which a feasible solution was found. Considering a time horizon of 60 days, the average gap for set-1 was 5.8%, while no feasible solutions were found for set-2. Our algorithm was able to find the same value of BKV for three instances (marked in bold). The local searches improved the objective function on average 11.4% considering only the results that did not use β_{jt} variables in the solutions obtained by the R&F. Moreover, they were able to remove the infeasibilities in five solutions found by the R&F. On the other hand, on average 85.4% of the total time was spent in the improvement phase. The average gap in relation to the lower bound was 37.2% for set-1 and 55.1% for set-2 (only considering feasible results).

Preliminary tests obtained new best-known values for two instances with $T = 45$, presented in Table 3. Column "CPU" presents the computer where the experiments were carried out, where "AMD" corresponds to the previously described computer, while "Intel" corresponds to an Intel Core i5-2300 running at 2.8 GHz, with 16 GB. Both experiments used the combination B of local search procedures. Also, they did not use auxiliary variables β_{jt}.

Table 3. New best-known-values found in preliminary experiments.

Instance	Parameters					R&F + LS			BKV	
	CPU	p	o	t_{rf}^{it}	t_{ls}^{it}	Time (s)	Obj	GAP$^{\text{BKV}}$	Time(s)	Obj
LR1_1_DR1_4_VC3_V11a	Core i5	5	20	50	50	1,578	-11,243	-0.03%	12,009	-11,239
LR1_1_DR1_4_VC3_V12b	AMD	5	15	50	20	1,942	-9,085	-0.17%	1,742	-9,069

Modified Instances Results. Considering the modified instances, the relative gap GAP^{LB} was on average 10.1% for set-1 and 32.5% for set-2, being smaller than the gap in the tests with MIRPLIB instances. This does not necessarily mean that our algorithm is better considering these instances, but the linear relaxations can be better in randomly distributed ports. Also, our algorithm was able to find more feasible solutions for the modified instances than the MIRPLIB instances. But, 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 Conclusion and Future Works

This work presented an extension of a relax-and-fix algorithm for solving a class of Maritime Inventory Routing Problem. Two sets of additional constraints were proposed, either for improving the bounds and for obtaining solutions faster. MIP-based local search procedures were used for improving the solutions and removing infeasibilities. Computational experiments were performed on MIR-PLIB and modified instances. Although it did not obtain feasible solutions for all instances, our algorithm found good solutions in reasonable time, including two best-known values for MIRPLIB instances. As future work, we intend to model MIRP as a fixed charge network flow as in [2], using valid inequalities for improving lower bounds, and using the relax-and-fix as the solution method.

Acknowledgements. The work of the first author was funded by CAPES - Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior.

References

1. Maritime inventory routing problem library (MIRPLIB), https://mirplib.scl.gatech.edu/, accessed: 2017-04-05
2. Agra, A., Andersson, H., Christiansen, M., Wolsey, L.: A maritime inventory routing problem: Discrete time formulations and valid inequalities. Networks 62(4), 297–314 (2013)
3. 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 pp. 1–25 (2016)
4. Christiansen, M.: Decomposition of a Combined Inventory and Time Constrained Ship Routing Problem. Transportation Science 33(1), 3–16 (1999)
5. Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D.: Ship routing and scheduling in the new millennium. European Journal of Operational Research 228(3), 467–483 (2013)
6. Christiansen, M., Fagerholt, K., Ronen, D.: Ship Routing and Scheduling: Status and Perspectives. Transportation Science 38(1), 1–18 (2004)
7. Goel, V., Furman, K.C., Song, J.H., El-Bakry, A.S.: Large neighborhood search for LNG inventory routing. Journal of Heuristics 18(6), 821–848 (2012)
8. Hewitt, M., Nemhauser, G., Savelsbergh, M., Song, J.H.: A branch-and-price guided search approach to maritime inventory routing. Computers and Operations Research 40(5), 1410–1419 (2013)
9. Jiang, Y., Grossmann, I.E.: Alternative mixed-integer linear programming models of a maritime inventory routing problem. Computers & Chemical Engineering 77, 147–161 (2015)
10. Papageorgiou, D.J., Cheon, M.S., Harwood, S., Trespalacios, F., Nemhauser, G.L., Stewart, H.M.: Recent Progress Using Matheuristics for Strategic Maritime Inventory Routing (2016)
11. 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 26(4), 825–847 (2014)
12. 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 235(2), 350–366 (2014)
13. Pochet, Y., Wolsey, L.A.: Production planning by mixed integer programming. Springer Science & Business Media (2006)
14. Song, J.H., Furman, K.C.: A maritime inventory routing problem: Practical approach. Computers & Operations Research 40(3), 657–665 (2013)
15. Uggen, K.T., Fodstad, M., Nørstebø, V.S.: Using and extending fix-and-relax to solve maritime inventory routing problems. TOP 21(2), 355–377 (2013)