

Publié par : Faculté des sciences de l'administration  
Published by: 2325, rue de la Terrasse  
Publicación de la: Pavillon Palasis-Prince, Université Laval  
Québec (Québec) Canada G1V 0A6  
Tél. Ph. Tel. : (418) 656-3644  
Télec. Fax : (418) 656-7047

Disponible sur Internet : <http://www4.fsa.ulaval.ca/la-recherche/publications/documents-de-travail/>  
Available on Internet  
Disponibile por Internet :

## **DOCUMENT DE TRAVAIL 2018-013**

Cyclic Maritime Routing of  
Platform Supply Vessels

Juliana Beatriz S. BOUCHER  
Leandro C. COELHO  
Virgílio J. M. FERREIRA FILHO  
Laura BAHIENSE

Document de travail également publié par le Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, sous le numéro CIRRELT-2018-18

**Mars 2018**

Dépôt legal – Bibliothèque et Archives nationales du Québec, 2018  
Bibliothèque et Archives Canada, 2018

ISBN 978-2-89524-470-7 (PDF)

# Cyclic Maritime Routing of Platform Supply Vessels

---

Juliana Beatriz S. Boucher<sup>1</sup>, Leandro C. Coelho<sup>2\*</sup>, Virgílio J. M. Ferreira Filho<sup>1</sup>,  
Laura Bahiense<sup>3</sup>

<sup>1</sup> Production Engineering Program, Federal University of Rio de Janeiro, Brazil

<sup>2</sup> Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Canada Research Chair in Integrated Logistics, 2325, rue de la Terrasse, Université Laval, Québec, Canada, G1V 0A6

<sup>3</sup> Systems Engineering and Computer Science Program, Federal University of Rio de Janeiro, Brazil

*\*Corresponding author: leandro.coelho@cirrelt.ca*

---

## ABSTRACT

High quality maritime petroleum transportation is critical to ensure the efficient and timely flow of goods, reducing total logistics costs and guaranteeing a smooth operation process. In this paper, we focus on the transportation of deck cargo to offshore units as observed in the operations of our industrial partner for replenishing its offshore units in Rio de Janeiro, Brazil. The main objective of this research is to determine the maritime routes to solve a cyclic routing problem, taking into account total travel time, opening hours at the offshore facilities, and imposed departure times at the port, among other operational constraints. We show that existing periodic routing models do not capture all the features of our problem. We describe the solution procedures currently used by the company, and formally formulate the problem mathematically. Given that the sizes of the instances are too big to be solved exactly, we propose an adaptive large neighborhood search algorithm to cope with large instances of the problem. The computational results indicate that our heuristic clearly outperforms the exact method and the solution currently in place, both in terms of solution quality, computational time, and size of the instances solved. Our solutions allow the company to operate with up to two fewer vessels, saving millions of dollars in fixed investment, or allowing the company to expand its operations under the current operational cost.

**Keywords:** OR in maritime industry, offshore logistics, maritime transportation.

**Acknowledgments:** This work was partly supported by the Emerging Leaders in the Americas Program (ELAP) from the government of Canada, Brazilian Research Council (CNPq), and by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant 2014-05764. These supports are gratefully acknowledged.

# 1 Introduction

The exploration and production operations of the petroleum industry are supported by a complex logistics system. Technological investments are essential for improving processes in the oil industry, thus, companies are looking for methods to increase their efficiency taking into account costs minimization and ensuring a high service level. Here we focus on the transportation of deck cargo to offshore units, for which high-quality transportation is critical to ensure the efficient and timely flow of products, helping reduce the total logistics cost.

Specifically, this work deals with maritime cargo transportation in the Campos Basin, one of the most important operations for the offshore exploration and production activities of oil in Brazil. According to the Brazilian National Agency of Petroleum, Natural Gas, and Biofuels (ANP), in May 2017, the production of oil in Brazil totaled 2.653 million barrels per day (bbl/d). The volume represents a growth of 4.5% in comparison with the previous month and of 6.7% compared to the same month in 2016 [1]. Oil production is expected to reach 5.2 million bbl/d in 2026, doubling the value registered in 2016 [12]. Indicators of a good service policy involve the reduction of travel costs and visits to maritime units (MUs), taking into account the total travel time and opening hours constraints of certain MUs.

Two of the main characteristics of the problem are the opening hours and the time between visits (cycle time) that must be respected in order to serve the MUs twice a week distributing its cargo demand uniformly among visits. Thus, given a port schedule for the vessels, it is necessary to build routes for which the supply sequence considers the MUs operating only during the day or night, in a way that the interval between visits does not exceed a time limit. Due to administrative strategies to facilitate vessel scheduling, the company seeks to keep the same MUs for two subsequent trips of a vessel. However, it is important to ensure that the vessel leaves the port following a given schedule and arrives at an installation which is opened or offers the lowest waiting time. Moreover, the cycle time in each MU should not be too long or too short in accordance with a weekly planning.

According to Kaiser & Snyder [20], “the study of the logistics of the upstream offshore oil and gas industry has been diverse, but not theoretically unified or well-developed”. These studies deal with topics such as operational research applied to offshore logistics systems, information management, outsourcing, and others. In fact, although many have gained an empirical understanding of this issue, academic knowledge is still in its early

stages. The existing approaches to route supply vessels in order to serve offshore units can be classified into several categories: involving pickup and delivery, heterogeneous fleet, multiple products, stochastic, demand and travel time priority of some cargo, multiple trips, weather issues, and restrictions such as the vessel deck area and time limits. In addition some problems can be directly related to the routing of supply vessels, to the number of berths at the port [23] and to offshore air transport [19].

Vessel routing and scheduling problems are different from those of other transportation modes because they operate under different conditions and require decision support models that fit the specific problem characteristics [5]. In general, as is done in problems with opening hours constraints [9, 32], it is necessary to add the complexity of opening hours resulting from the fact that some customers impose delivery deadlines and earliest-delivery-time constraints. Thus, in addition to the total travel distance, the objective must include a penalty for waiting when a vessel arrives too early at a customer location.

The work plan for cyclic maritime routing is similar to that of periodic vehicle routing problem (PVRP), including other complicating constraints related to the port scheduling, opening hours and time penalties. Hence, no existing algorithm can be trivially adapted to solve the problem at hand. The PVRP is a generalization of the vehicle routing problem (VRP) in which vehicle routes are constructed for a period of time and was introduced by Beltrami & Bodin [2] for assigning hoist compactor trucks in municipal waste collection and has evolved into a significant body of work with several variants and applications arising in recent years [14]. They propose heuristics to solve the problem, but do not formally formulate or define it; however, they do include a good exposition of the difficulty of the PVRP, as compared to the standard VRP. A formal definition of the PVRP is provided by Russell & Igo [27] and they examine the difficulties of choosing a schedule for each node together with solving the routing problem, and consider a mixed integer problem, imposing constraints on the vehicle capacity as well as the maximum duration of the routes.

The first formulation of the PVRP is presented by Christofides & Beasley [6] and define it as the problem of designing a set of routes for each day of a given planning period to meet the required customer visit frequency. The integer programming formulation uses decision variables for the routing of a given vehicle on a given day and for the assignment of customers to schedules. Francis et al. [13] develop an exact solution method based on the Lagrangian relaxation of an integer programming formulation of the PVRP and solve a tactical version, and Mourgaya & Vanderbeck [22] solve a version in which visit schedules and customer assignment to vehicles are solved simultaneously. But when it

comes to real-life, most of the studies rely on heuristics or metaheuristics algorithms.

There are different kinds of heuristic methods to solve the PVRP. Similar methods to that of Beltrami & Bodin [2] working in two phases, are commonly found in heuristics for this problem [27, 15, 33]. Metaheuristic methods for this problem include those of Chao et al. [3], Cordeau et al. [10] and Drummond et al. [11].

One heuristic that can be applied to a large class of difficult optimization problems is the adaptive large neighborhood search (ALNS) [24]. The ALNS framework is based on the large neighborhood search (LNS) algorithm presented by Shaw [29], in which a number of competing sub-heuristics are chosen in order to modify the current solution. An algorithm is chosen to destroy the current solution at each iteration, followed by another one to repair the partial solution. Finally, if the new solution satisfies some criteria defined by the local search framework it is accepted. ALNS can be based on any local search framework as simulated annealing, tabu search and guided local search, for example. Several problems related to vehicle routing have been solved through ALNS obtaining positive results [24, 8, 18, 26].

However, there are few studies on offshore logistics systems treating the PVRP, as observed by Halvorsen-Weare et al. [17] and Shyshou et al. [30]. Shyshou et al. [30] present an LNS heuristic for the periodic supply vessel planning problem and fleet composition problem. Halvorsen-Weare et al. [17] develop a voyage-based solution method that consists of two phases: the first one generates candidate voyages the vessels may perform and while the second solves the voyage-based model.

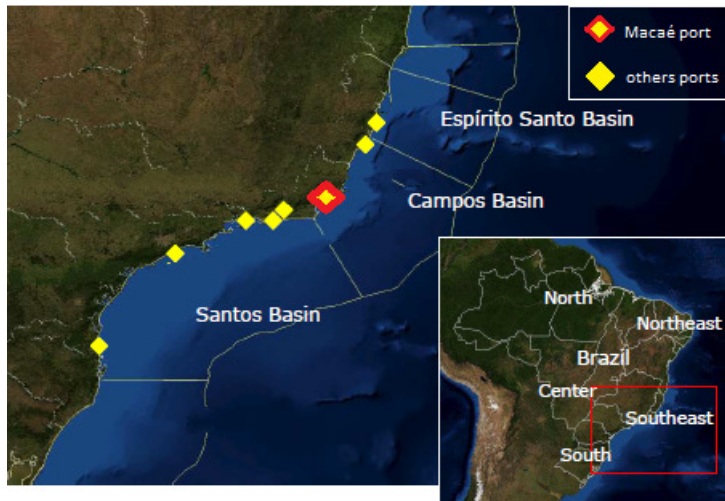
In this paper we describe, model and solve the operations involving the transportation of deck cargo to the offshore units defining the routes to support MUs. The great difficulty of solving this PVRP is to serve the customers taking into account availability of service, travel time limit, cycle time to serve them twice in a week, and flexible departure times at the port. We consider real-life large-scale instances, containing up to 61 installations, in contrast to others in the literature who solved the problem with up to 31 installations [30]. In order to solve it efficiently, we propose an ALNS heuristic capable of providing high-quality solutions within short computational time.

The remainder of this paper is organized as follows. In Section 2 we provide a formal description of the problem, followed by a mathematical formulation in Section 3. Our heuristic is introduced in Section 4 where we describe an initial solution procedure and many operators specifically designed to solve the problem at hand. The details of extensive computational experiments are described in Section 5. Our conclusions follow in Section

## 2 Problem description

Maritime exploration and production operations are supported by a complex logistics system and service, called maritime support. This system consists of ports, airports, warehouses, helicopters and uses several specialized vessels to transport cargo for MUs. The logistics system to support the exploration and production operations is primarily cargo transport for maritime units and return cargo, storage of materials and equipment at the port, and people transportation.

The costs of shipping operation and the value of oil production are very high, especially the rent of the drillships that have an average daily rate of US\$500,000 [31]. Thus, it is necessary to avoid disruption of these activities due to delays in cargo transportation. This highlights the importance of planning for minimizing costs of these operations. In the case of our industrial partner, the offshore basins located in the southeast of Brazil are Santos, Espírito Santo, and Campos as shown in Figure 1. The Campos basin represents about 80% of the Brazilian production of oil and gas being mainly served by the port of Macaé, which is the main port for maritime support in Brazil dealing with more than 50% of deck cargo flow, located 180 km north of Rio de Janeiro [21].



**Figure 1:** Map of the southeast of Brazil, including the port configuration to support the logistics activities (Adapted from Leite [21])

The main customers are production units, drilling rigs and special vessels. While production units have a fixed location, drilling rigs, and special vessels have variable location

depending on wells where they meet at a certain time. The vessels used are platform supply vessels (PSVs), specialized in support of drilling or production platforms. Their main function is to transport supplies to platforms and usually return with loads to the coast.

The problem lies in determining the schedules for the routes. In this setting, a route is a group of customers that must be served in two trips over a planning horizon, typically a week, with each trip starting and ending at the supply depot. The demands of installations are given in  $m^2$  of deck capacity and they are calculated dividing their weekly demand by the number of visits. The total capacity is limited to the maximum deck area available for the homogeneous fleet. There are also backloads that need to be transported from the offshore installations to the onshore supply depot.

Docking and loading times are also given. Operations at the port and at MUs are computed in terms of lift units, describing the number of crane movements required to move cargo from and to vessels. Moreover, offshore facilities operate based on opening hours, when they are available to be served by the vessels, and a port schedule must be respected as well.

Due to administrative strategies to facilitate the vessel scheduling, the company must assign the same MUs in both trips of a vessel. However, the time between weekly visits to each MU, referred to as cycle time should not be too long or too short.

In order to avoid major changes in routine operation, in general, drilling and production units are routed in separate groups. The production units are separated into the operational units of Campos (UB) and Rio de Janeiro (UR), while drilling units are routed separately. Data obtained from our partner includes demands, service time and opening hours, routes currently performed by the company, current clusters of maritime units, the location of units, and port scheduling.

### 3 Mathematical model

This section presents the mathematical model, that is defined on a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$  where  $\mathcal{V}$  is the node set and  $\mathcal{A}$  corresponds to the set of arcs. The node set  $\mathcal{V}$  is defined as  $\mathcal{V} = \{0\} \cup \mathcal{C} \cup \mathcal{S}$ , where node 0 represents the port and set  $\mathcal{C} = \mathcal{P} \cup \mathcal{B}$  represents the maritime units, that are artificially duplicated and partitioned into two sets:  $\mathcal{P} = \{\text{units visited at the first travel}\}$  and  $\mathcal{B} = \{\text{the same units served at the second travel}\}$ . We also define  $\mathcal{T} \subseteq \mathcal{C}$  as the set of units having opening hours constraints, in such a way that

$\mathcal{T} = \mathcal{D} \cup \mathcal{N}$ , where  $\mathcal{D} = \{\text{the set of units to be served in daytime}\}$  and  $\mathcal{N} = \{\text{the set of units to be served in night time}\}$ . Finally, in order to allow different departure times from the port 0, we create a set  $\mathcal{S}$  of auxiliary nodes, such that  $\mathcal{S} = \mathcal{S}^1 \cup \mathcal{S}^2$ , where  $\mathcal{S}^1$  contains the nodes corresponding to the first travels and  $\mathcal{S}^2$  contains the nodes corresponding to the second travels. The arc set  $\mathcal{A}$  contains all pair of nodes  $(i, j)$  such that  $i \neq j$ ,  $i, j \in \mathcal{V}$ , and for each arc  $(i, j)$  non-negative travel distances  $d_{ij}$  and travel times  $t_{ij}$  are known. The problem is defined over a finite time horizon, typically a week, on the set  $\mathcal{J}$  of days.

Each unit  $i \in \mathcal{C}$  has associated: a service time  $s_i$ , lifts  $z_i$ , demand  $k_i$ , and backload  $\rho_i$ . Additionally, to each unit  $i \in \mathcal{T}$  is related an opening hours interval  $[a_i^d, b_i^d]$  for  $d \in \mathcal{J}$ . A vessel needs to leave the port 0 at the scheduled departure time  $\lambda_0^d$  for  $d \in \mathcal{J}$  and is allowed to arrive at the unit before its time window opens, but it has to wait until the start of the time window for the service to begin. During a week, each unit must be visited twice, then we consider that a cycle time between first and second visit needs to lie in the range  $[n_{min}, n_{max}]$ .  $R$  is a maximum travel time,  $H$  represents the maximum berthing time,  $\alpha$  and  $\beta$  are coefficients of the regression used to determine berthing time, and  $M$  is a big number. The maximum capacity of the vessels is denoted  $\delta$ . Finally,  $\mu$  is a weight used in the objective function to balance its second term according to the magnitude of the sum of the distances.

Seven types of variables are used in the mathematical model:  $x_{ij}$ ,  $i, j \in \mathcal{V}$  is a binary variable taking value one if the arc between nodes  $i$  and  $j$  is used, zero otherwise;  $y_i^d$ ,  $i \in \mathcal{T}$ ,  $d \in \mathcal{J}$  is a binary variable which is one if customer  $i$  is served on period  $d$ , zero otherwise;  $f_{ij}$  and  $g_{ij}$  are non-negative real variables representing, respectively, demand and lifts flows passing through arc  $(i, j)$  for  $i \in \mathcal{V}$  and  $j \in \mathcal{C} \cup \mathcal{S}$ ; the waiting times are represented by  $w_i^1$  for  $i \in \mathcal{D}$  and they are added to the objective function whenever the range is violated; and finally,  $h_i^1$  and  $h_i^2$  are non-negative real variables that indicate the arrival times at node  $i \in \mathcal{C} \cup \mathcal{S}$  considering the port departure times and setting them to zero in order to obtain the total travel time, respectively. The mathematical model is stated as follows:

$$\min \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} d_{ij} x_{ij} + \mu \sum_{i \in \mathcal{T}} w_i^1 \quad (1)$$

**s.t.**

$$\sum_{j \in \mathcal{C}} x_{ij} = 1 \quad i \in \mathcal{C} \cup \mathcal{S} \quad (2)$$

$$\sum_{i \in \mathcal{C}} x_{ij} = 1 \quad j \in \mathcal{C} \cup \mathcal{S} \quad (3)$$

$$x_{ij} = 0 \quad i \in \mathcal{C}, j \in \mathcal{S} \quad (4)$$

$$x_{ij} = 0 \quad i \in \mathcal{P}, j \in \mathcal{B} \quad (5)$$

$$\sum_{j \in \mathcal{S}} x_{0j} = \sum_{i \in \mathcal{C}} x_{i0} \quad (6)$$

$$f_{ij} \leq \delta x_{ij} \quad i, j \in \mathcal{V} \quad (7)$$

$$g_{ij} \leq M x_{ij} \quad i, j \in \mathcal{V} \quad (8)$$

$$f_{ij} \leq (\delta - \rho_j) x_{ij} \quad i \in \mathcal{S}, j \in \mathcal{C} \quad (9)$$

$$\sum_{i \in \mathcal{V}} f_{ij} - \sum_{i \in \mathcal{V}} f_{ji} = k_j \quad j \in \mathcal{C} \quad (10)$$

$$\sum_{i \in \mathcal{V}} g_{ij} - \sum_{i \in \mathcal{V}} g_{ji} = z_j \quad j \in \mathcal{C} \quad (11)$$

$$f_{0i} = f_{0j} \quad i = j, i \in \mathcal{S}^1, j \in \mathcal{S}^2 \quad (12)$$

$$g_{0i} = g_{0j} \quad i = j, i \in \mathcal{S}^1, j \in \mathcal{S}^2 \quad (13)$$

$$h_i^1 + w_i^1 + s_i + t_{ij} - M(1 - x_{ij}) \leq h_j^1 \quad i, j \in \mathcal{C} \quad (14)$$

$$h_i^2 + w_i^1 + s_i + t_{ij} - M(1 - x_{ij}) \leq h_j^2 \quad i, j \in \mathcal{C} \quad (15)$$

$$h_j^2 + s_j + t_{j0} - M(1 - x_{j0}) \leq R \quad j \in \mathcal{C} \quad (16)$$

$$\alpha(g_{0i}) + \beta \leq H \quad i \in \mathcal{S} \quad (17)$$

$$\sum_{d \in \mathcal{J}} y_i^d = 1 \quad i \in \mathcal{D} \quad (18)$$

$$\sum_{d \in \mathcal{J}} y_i^d = 1 \quad i \in \mathcal{N} \quad (19)$$

$$h_i^1 = \lambda_0^d \quad i \in \mathcal{S}, d \in \mathcal{J} \quad (20)$$

$$h_i^2 = 0 \quad i \in \mathcal{S} \quad (21)$$

$$n_{min} \leq h_j^1 - h_i^1 \leq n_{max} \quad i = j, i \in \mathcal{P}, j \in \mathcal{B} \quad (22)$$

$$a_i^d y_i^d \leq h_i^1 \quad i \in \mathcal{T}, d \in \mathcal{J} \quad (23)$$

$$h_i^1 + s_i \leq b_i^d y_i^d + M(1 - y_i^d) \quad i \in \mathcal{T}, d \in \mathcal{J} \quad (24)$$

$$w_i^1 = 0 \quad i \notin \mathcal{T} \quad (25)$$

$$x_{ij}, y_i^d \in \{0, 1\} \quad i, j \in \mathcal{V}, d \in \mathcal{J} \quad (26)$$

$$h_i^1, h_i^2, f_{ij}, g_{ij}, w_i^1 \geq 0 \quad i, j \in \mathcal{V} \quad (27)$$

The first term of the objective function (1) minimizes the sum of the traveled distances, while the second minimizes the weighted sum of the waiting times.

Assignment constraints (2) and (3) ensure that there is only one incoming and one outgoing arc for each customer. Constraints (4) guarantee that artificial departure nodes are attended first. Constraints (5) assure that there are only arcs between customers from the same travel. Constraint (6) establishes that the number of arcs leaving and entering

the port must be the same. Constraints (7) and (8) define upper bounds for the demands and lifts flows, respectively. Constraints (9) ensure free space on the vessels for their first backloads. Constraints (10) and (11) define the demand and lift flow for each customer, respectively. Constraints (12) and (13) link pairs of port departure times for first and second travels through the demands and lifts flows representing the same customer in each travel. Constraints (14) and (15) define the relation between arrival times from a customer and its successor respecting the port departure times on the auxiliary nodes and setting them to zero, respectively, including the waiting times when they are necessary to respect the opening hours. Constraints (16) guarantee that the routes have a maximum travel time from the port departure time set to zero to the return of the vessel to port. Constraints (17) limit the berthing time in order to avoid delays in port schedule. The choice of a day of the week to serve daytime and night-time customers are ensured by constraints (18) and (19), respectively. Constraints (20) and (21) assure that the arrival time to the auxiliary nodes should be equal to the beginning of time window given by the port schedule and setting to zero, respectively. Constraints (22) guarantee that the cycle time between first and second travels must lie within a given interval. Constraints (23) and (24) ensure that the service must be realized within day or night operation period. Constraints (14), (16) and (24) were linearized by introducing a sufficiently large number  $M$ . Constraints (25) define null waiting times for that nodes not belonging to the set of units having opening hours constraints. Finally, constraints (26) and (27) enforce the domain of the variables.

## 4 Adaptive large neighborhood search heuristic

Due to the large size of the instances, the techniques to obtain an exact optimal solution of the complete problem in a timely manner are not efficient. In the other words, the mixed integer linear programming (MILP) modeling for the PRVP is very difficult to solve over real data. Therefore, according to Christiansen et al. [4] the search for the solution of more realistic problems are facilitated by the development of fast optimization algorithms and the use of heuristic methods and/or metaheuristic replacing or associated with exact algorithms able to reach to the near optimal solution in less time consuming and computational effort.

The proposed heuristic was introduced by Ropke & Pisinger [26] and extends the LNS heuristic of Shaw [29]. The LNS heuristic has been applied to the PVRP in specific supply vessel problems [30] and has achieved positive results. Usually, LNS algorithms

are built on neighborhood moves that make small changes to the current solution. These methods are able to investigate numerous solutions in a short time, but when applied to very constrained problems can have difficulties in moving from one promising area of the solution space to another [26].

ALNS is a local search framework where a large collection of variables are modified in each iteration through some simple operators selected independently. The current solution is partially destroyed and then repaired using destroy and repair operators chosen randomly and the new solution is accepted if it satisfies some predefined criteria. Alternatively, we can see this adaptive heuristic as a sequence of fix and optimize operations where a number of variables are fixed at their current value and the optimize operation seeks to find a near-optimal solution that respects the fixed variables and after all variables are unlocked again. This is an interesting point of view to problems that the destroy and repair operations do not seem intuitive [24].

ALNS has already been adapted to several transportation problems including node routing [26], arc routing [28], cumulative capacitated VRP [25] and inventory-routing [7]. This heuristic can be based on different criteria such as, e.g., simulated annealing, tabu search, guided local search and a simple acceptance criteria would be to accept all improving solutions. In this project, we use an acceptance criterion based on simulated annealing.

A roulette wheel mechanism controls the choice of operators with a probability that depends on their past performance. To each operator  $i$  are associate a score  $\lambda_i$  and a weight  $w_i$ . Then, operator  $j$  is selected with probability  $w_j / \sum_{i=1}^h w_i$ ,  $h$  being the number of operators. The weights are computed according to the performance operator during a segment of iteration  $\varphi$ . The score of the selected operator is increased at each iteration by  $\lambda_1$  if the operator finds a new best solution, by  $\lambda_2$  if it finds a solution better than the incumbent, and by  $\lambda_3$  if the solution is not better but still accepted. After  $\varphi$  iterations, finally considering the scores obtained in the last segment the weights are updated. Then, the scores are reset to zero at the end of each segment.

Formally, let  $z(s)$  be the solution cost and  $\tau > 0$  be the current temperature. Then, given a current solution  $s$ , a neighbor incumbent solution  $s'$  is always accepted if  $z(s') < z(s)$ , and is accepted with probability  $e^{-(z(s')-z(s))/\tau}$  otherwise. The temperature starts at  $\tau_0$  and is decreased by a cooling rate factor  $\gamma$  at each iteration, where  $0 < \gamma < 1$ .

## 4.1 Initial solution

The algorithm is initialized with a solution obtained through the sweep algorithm based in Gillett & Miller [16]. As the problem is Euclidean, the locations of customers can be represented by their polar coordinates  $(\theta_i, \rho_i)$  with origin at the port. The vertices are numbered in increasing order of the angle between the line linking the vertex to a reference point (port). After that, the customers are chosen in ascending order of their angles. The implementation takes place as follows: we first choose a vessel not used, then assign MUs to the vessel from an MU not yet routed with the smallest angle given that its capacity and total travel time are not exceeded. We then repeat the process until all customers have been routed.

## 4.2 Destroy operators

After finding an initial solution we remove and insert customers using the destroy and repair operators. The number of customers that are removed in a route is chosen randomly bonded by  $[1, n]$  where  $n$  is a random number following a semi-triangular distribution with a negative slope. There is a major transformation of the solution when  $n$  is large, however, it is necessary choosing more often a small number  $n$ . Due to a high number of combinations, if  $n$  and the total number of customers are large, the computational time will be large as well. We now list the destroy operator we have developed.

1. *Randomly remove  $n$  customers:* this simple removal heuristic removes randomly the  $n$  customers from the first and second trips in the current solution. The main characteristic of this operator is diversifying the search when it is stuck in a local minimum.
2. *Remove  $n$  customers with the largest distance saving:* this operator identifies which customer represents the largest total distance saved by the vessel and it is removed. The process continues until all  $n$  selected customers are removed. This heuristic aims to save computational time when it calculates only a part of the objective function.
3. *Remove  $n$  customers with the largest time saving:* this operator removes the customers that represent the largest time saving when they are not considered in the solution. The time variables are the waiting that is minimized in the objective function.

4. *Remove  $n$  customers with the smallest total cost:* this operator removes the customer that represents the lowest value of the objective function when they are not considered in the solution. Hence, it takes more computational time to solve the problem because it considers all the time variables and total distance.

### 4.3 Repair operators

After using destroy operators, the same number of customers removed from the solution must be reinserted following a repair operator. It is important to emphasize that each customer is inserted into different positions but at the same route of both first and second travels. The feasibility criteria are maximum vessel capacity, total travel and berthing times. The list of repair operators follows:

1. *Randomly insert customers:* the customers previously removed, are now reinserted in random routes and positions, ensuring feasibility. Similar to the random removal operator, its function is to diversify the search.
2. *Best insertion:* this operator evaluates each customer and inserts the one with smallest increase to the objective function. This process is repeated until all removed customers have been inserted.
3. *Regret-2:* the regret heuristic chooses the insertion with the biggest regret if it is not done immediately. This operator tries to improve the solution by incorporating a kind of look ahead information when selecting the request to insert. The regret value is the difference in the cost of inserting the request in its best route and its second best route. This operator is time consuming but it performs a more thorough search.

### 4.4 Parameters settings and pseudocode

In our implementation we have used the following parameters settings. The starting and ending temperature are, respectively,  $\tau_0 = 30000$  and  $\tau_{final} = 0.01$ . The cooling rate  $\gamma = (0.01/\tau_0)^{1/i_{max}}$ , where  $i_{max}$  is the maximum number of iterations, is a function of the desired number of iterations, adjusting according to the probability that the ALNS mechanism accepts worse solutions. The stopping criterion is satisfied when  $i_{max}$  iterations have been performed.

In order to assign a random variability process when the ALNS heuristic gets stuck in a local optimum, a reheating is activated if the number of non improving solutions  $i_{noimprove}$  reaches  $r$ . According to our performed tests, this limit is set to  $r = 1000$  iterations. As shown by Pisinger & Ropke [24], it is not needed to remove a large number of customers in the removal phase; then the parameter of maximum number of clients removed is  $n = 5$ .

In our implementation, the maximum number of iterations  $i_{max}$  was set to 30000 for instances with fewer than 25 clients, 15000 for instances with fewer than 45 clients and 8000 for instances with up to 60 clients. The segment length is  $\varphi = 100$  and the reaction factor  $\eta$  was set to 0.7, thus defining the new weights by 70% of the performance on the last segment and 30% of last weight value. The scores are updated with  $\lambda_1 = 10$ ,  $\lambda_2 = 5$  and  $\lambda_3 = 2$ . Algorithm 1 shows the pseudocode of ALNS heuristic.

---

### Algorithm 1 ALNS Heuristic

---

```

1: Initialize: set all weights equal to 1 and all scores equal to zero.
    $s_{best} \leftarrow s \leftarrow$  initial solution,  $\tau \leftarrow \tau_0$ .
2: while  $i < i_{max}$  do
3:    $s' \leftarrow s$ ;
   Apply a destroy and a repair operator to  $s'$  and update the number of times they are used.
4:   if  $z(s') < z(s)$  then
5:      $s \leftarrow s'$ ;
6:      $i_{noimprove} \leftarrow 0$ 
7:     if  $z(s) < z(s_{best})$  then
8:        $s_{best} \leftarrow s$ ;
       update the score for the used operators with  $\lambda_1$ ;
9:     else
10:      update the score for the used operators with  $\lambda_2$ ;
11:    end if
12:  else
13:    if  $s'$  is accepted by simulated annealing criterion then
14:       $s \leftarrow s'$ ;
      update the score for used the operators with  $\lambda_3$ ;
15:       $i_{noimprove} \leftarrow i_{noimprove} + 1$ ;
16:    end if
17:  end if
18:  if iteration count is a multiple of  $\varphi$  then
19:    update weights of all operators and reset their scores.
20:  end if
21:   $\tau \leftarrow \phi\tau$ ;
22:   $i \leftarrow i + 1$ ;
23:  if  $i_{noimprove} \geq r$  then
24:     $\tau = \tau_0$ ;
25:     $i_{noimprove} \leftarrow 0$ ;
26:  end if
27: end while
28: return  $s_{best}$ ;

```

---

## 5 Computational experiments

In this section we present the results of detailed computational experiments carried out to access the performance of our algorithm on instances provided by our industrial partner. The company currently employs a solution method based on a heuristic clustering followed by exact method. Our algorithm is coded in C++ and run on computers equipped with Intel Xeon 5450 processors running at 3.0 GHz with up to 32Gb of RAM with the Scientific Linux 3.13 operating system.

In Section 5.1 we describe the instances and benchmark received from the company, followed by the description of detailed computational results in Section 5.2 and by a sensitivity analysis in Section 5.3.

### 5.1 Instances

The datasets obtained from the company are divided into two classes: UR (MUs from Rio de Janeiro) and UC (MUs from Campos). This division between UR and UC is administrative and it is used to cluster old and new oil platforms; we propose treating all customers at the same time while obtaining good results in a short computing time. We have then created larger instances by combining these two datasets, referred to as UR + UC.

The size of these instances varies over time following several factors such as the need for more maintenance or if MUs are not operational for a period of time. In general, UR has 18 to 21 MUs, UC has 40 to 42 MUs and the combined instance has 59 to 61 MUs. There are two customers with service constraints during the day and one during the night. The maximum capacity considered is 600 m<sup>2</sup>. The length of a period (day and night) lasts 12 hours. The instances are named taking into account the kind of cluster and the year in which they were programmed.

Since there is an integration between port and routing scheduling the company opted to develop a heuristic with three steps: a clustering heuristic to divide problem into smaller ones, an exact method to solve the VRP with capacity and travel time constraints on each cluster and finally, given a port scheduling, rebuilding the sequence of routes containing customers with opening hours and cycle time constraints using an exact method again. Therefore, the company solution is very limited and spends a lot of time to solve each instance in a sequential fashion.

## 5.2 Computational results

We have implemented the model presented in Section 3 using a state of the art commercial solver. After running it for two hours on each instance, the solver was not capable of finding a feasible solution, and the lower bounds were consistently low and far from the solution obtained by the company.

We start the presentation of our results by comparing the solution obtained by the company alongside our solution for the clustered problem. In Table 1 we show the clustered instances followed by the number of MUs considered and the cost of the solution employed by the company. Next to that we show the solution cost of our ALNS algorithm (best and average over 10 runs), the average computational time and the percentage improvement over the company solution. We note that the solution obtained by the company takes about one hour of computing time, compared to no more than 10 minutes for our algorithm. Moreover, our method consistently outperforms that of the company, obtaining improvements of over 11%.

Instance	$n$	Company solution	ALNS			
			Best	Avg	Time (s)	Improv (%)
<i>UR</i> – 1	21	3259.37	3248.24	3248.89	535	0.34
<i>UR</i> – 2	18	2523.77	2521.92	2522.20	405	0.07
<i>UR</i> – 3	18	3215.72	3183.01	3183.32	520	1.02
<i>UR</i> – 4	19	3325.74	3290.19	3300.45	450	1.07
<i>UC</i> – 1	40	4840.87	4305.5	4405.97	520	11.06
<i>UC</i> – 2	41	4773.37	4300.74	4367.50	560	9.90
<i>UC</i> – 3	42	5478.4	4870.21	5150.51	535	11.10
<i>UC</i> – 4	40	5380.18	5035.43	5303.65	460	6.41
<b>Average</b>		4099.67	3844.40	3935.31	498	5.12

**Table 1:** Summary of results, % improvement and computational times for instances UR and UC

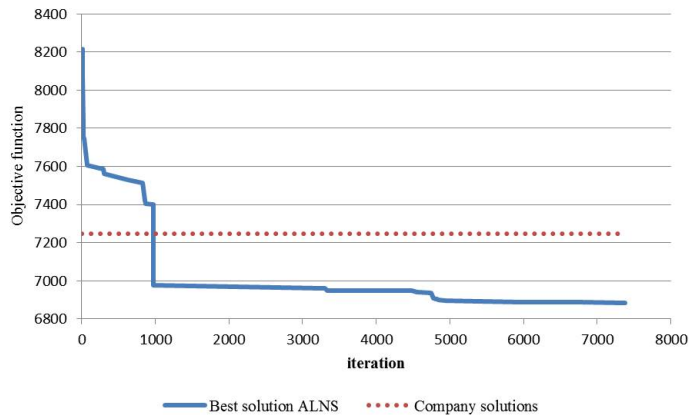
When all MUs are considered together, the solution of the company is simply to combine both individual solutions. In Table 2 we show how we have handled the instances of class UR+UC. The table shows the instance name followed by the number of MUs, the solution of the company and the sum of our individual solutions. We then show our integrated solution considering all MUs (best and average), followed by the running time and improvement over the solution of the company.

It is obvious that our clustered solutions were already better than those of the company, but our integrated solutions can find synergies and improvements when scheduling and routing the vessels, further improving the solutions for this difficult and big problem.

Instance	$n$	Compay solutions	ALNS sum of clustered solutions	ALNS			
				Best	Avg	Time (s)	Improv (%)
$UR + UC - 1$	61	8100.24	7553.74	7331.46	7493.45	650	9.19
$UR + UC - 2$	59	7297.14	6822.66	6750.35	6863.95	586	6.84
$UR + UC - 3$	60	8694.12	8053.22	7909.53	8043.66	572	9.02
$UR + UC - 4$	59	8705.92	8325.62	8197.28	8384.20	485	5.85
<b>Average</b>		8179.85	7688.81	7547.15	7696.31	573.25	7.72

**Table 2:** Summary of results, % improvement and computational times for instances UR+UC

In Figure 2 we depict the evolution of the solution cost with respect to the number of iterations, and we observe that after about 1,000 iterations our algorithm yielded a solution better than that of the company. This solution was still significantly improved in the later phases of our metaheuristic.



**Figure 2:** Best solutions: company vs ALNS for the instance  $UR + UC - 2$

Additionally, as shown in Table 3, our solutions sometimes require fewer vessels than those of the company. This is extremely important since these vessels can cost several million dollars, and being able to perform the required service with fewer of vessels can save the company a significant amount of money, or allow the company to grow and expand its operations without huge fixed investments on new vessels. Moreover, this result shows that the company should reconsider its administrative decision to cluster the MUs as they are now.

Instance	# of vessels	
	Company	ALNS
$UR - 1$	4	4
$UR - 2$	3	3
$UR - 3$	4	4
$UR - 4$	4	4
$UC - 1$	7	6
$UC - 2$	7	6
$UC - 3$	9	8
$UC - 4$	8	7
$UR + UC - 1$	11	10
$UR + UC - 2$	10	9
$UR + UC - 3$	13	11
$UR + UC - 4$	12	11

**Table 3:** Comparative of number of voyages per instance

### 5.3 Sensitive analyses

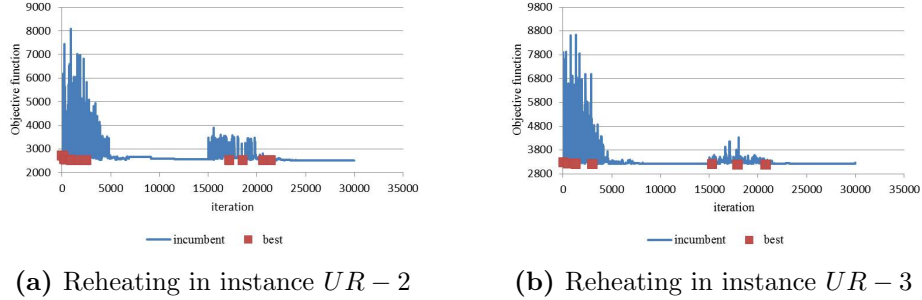
In this section we evaluate the stability and performance of the algorithm with respect to some of its parameters. We start by presenting an analysis using the coefficient of variation, which represents the ratio of the standard deviation to the mean and is computed as  $Dev = (Standard\ deviation / Mean)$ , applying the heuristic ten times for each instance. The results are presented in Table 4 and it can be considered with a low-variance.

Instance	Dev(%)
$UR - 1$	0.01
$UR - 2$	0.03
$UR - 3$	0.01
$UR - 4$	0.12
$UC - 1$	1.21
$UC - 2$	1.01
$UC - 3$	3.33
$UC - 4$	2.30
$UR + UC - 1$	1.23
$UR + UC - 2$	0.90
$UR + UC - 3$	1.05
$UR + UC - 4$	1.85
<b>Average</b>	1.09

**Table 4:** Coefficient of variation

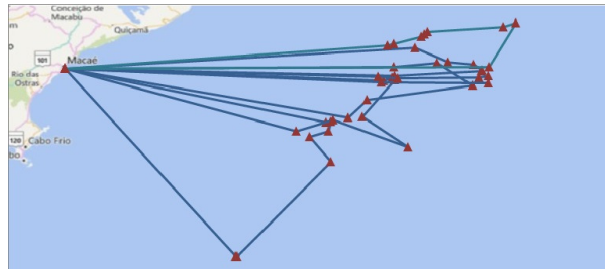
In order to apply a diversification and intensification of the search, we consider one reheating if the algorithm reaches  $n$  iterations without improvement, thus the temperature is set to  $\tau_0$  and the process is repeated. The graph of the Figure 3 clearly shows the efficiency of the reheating in small instances. After this procedure, the algorithm accepts

the worst solutions again, since the temperature is high, and finds other neighborhood generating new best solutions.

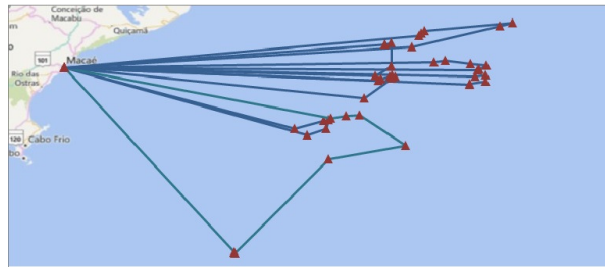


**Figure 3:** Reheating in small instances

Finally, we depict a graphical representation of the routes along the coast of Brazil. In Figure 4a we show the routes of the company solution, and in Figure 4b the routes of our solution. It can be seen that the change in the shape of the solution is significant, once again reinforcing the idea that our solution improved not only the routing aspect of the problem but also its scheduling and clustering of MUs in each route.



(a) Company routes (solution cost = 4773.37)



(b) ALNS routes (solution cost = 4300.74)

**Figure 4:** Routes map

## 6 Conclusion

We have solved a cyclic maritime routing problem with the objective of minimizing the distances and wait times. We have proposed a mixed-integer programming formulation but it did not obtain a feasible solution after two hours of computing time. We have then developed an ALNS algorithm with several operators specific to the problem at hand. Through computational experiments based on real-world instances, we have compared our ALNS against the solutions currently used by the company. All instances studied were significantly improved. We have reduced the average distance by 6.98% using only half the company execution time, and reduced the number of supply vessels used by creating a schedule with two fewer routes to some instances. Our computational results indicate that our heuristic clearly outperforms the solutions given by the heuristic solution currently in place, in terms of cost minimization, computational time, and size of the instances solved.

## References

- [1] ANP (2017). National Agency of Petroleum, Natural Gas and Biofuels, Oil production up 4.5% in May over April. URL: <http://www.anp.gov.br/wwwanp/noticias/anp-e-p/3844-producao-de-petroleo-aumentou-4-5-em-maio-em-relacao-a-abril> accessed March 6, 2018.
- [2] Beltrami, E., & Bodin, L. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, *4*, 65–94.
- [3] Chao, I., Golden, B. L., & Wasil, E. A. (1995). An improved heuristic for the period vehicle routing problem. *Networks*, *26*, 25–44.
- [4] Christiansen, M., Fagerholt, K., Nygreen, B., & Ronen, D. (2007). Chapter 4 maritime transportation. In C. Barnhart, & G. Laporte (Eds.), *Transportation* (pp. 189 – 284). Elsevier volume 14 of *Handbooks in Operations Research and Management Science*.
- [5] Christiansen, M., Fagerholt, K., & Ronen, D. (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science*, *38*, 1–18.
- [6] Christofides, N., & Beasley, J. (1984). The period routing problem. *Networks*, *14*, 237–256.

- [7] Coelho, L., Cordeau, J.-F., & Laporte, G. (2012). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, *24*, 270–287.
- [8] Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). The inventory-routing problem with transshipment. *Computers & Operations Research*, *39*, 2537–2548.
- [9] Coelho, L. C., Renaud, J., & Laporte, G. (2016). Road-based goods transportation: a survey of real-world logistics applications from 2000 to 2015. *INFOR: Information Systems and Operational Research*, *54*, 79–96.
- [10] Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, *30*, 105–119.
- [11] Drummond, L., Ochi, L., & Vianna, D. (2001). An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Generation Computer Systems*, *17*, 379–386.
- [12] EPE (2017). Energy Research Company, Oil Production Forecast. URL: <http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-40/PDE2026.pdf#search=PDE2026%20cap%C3%ADtulo%205> accessed March 6, 2018.
- [13] Francis, P., Smilowitz, K., & Tzur, M. (2006). The period vehicle routing problem with service choice. *Transportation Science*, *40*, 439–454.
- [14] Francis, P. M., Smilowitz, K. R., & Tzur, M. (2008). The period vehicle routing problem and its extensions. In B. L. Golden, S. Raghavan, & E. A. Wasil (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 73–102). Boston, MA: Springer US.
- [15] Gaudioso, M., & Paletta, G. (1992). A heuristic for the periodic vehicle routing problem. *Transportation Science*, *26*, 86–92.
- [16] Gillett, B., & Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, *22*, 340–349.
- [17] Halvorsen-Weare, E., Fagerholt, K., Nons, L., & Asbjørnslett, B. (2012). Optimal fleet composition and periodic routing of offshore supply vessels. *European Journal of Operational Research*, *223*, 508 – 517.

- [18] Hansen, P., & Mladenović, N. (1999). An introduction to variable neighborhood search. In S. Voß, S. Martello, I. H. Osman, & C. Roucairol (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* (pp. 433–458). Boston, MA: Springer US.
- [19] Hermeto, N., Ferreira Filho, V., & Bahiense, L. (2014). Logistics network planning for offshore air transport of oil rig crews. *Computers & Industrial Engineering*, *75*, 41–54.
- [20] Kaiser, M., & Snyder, B. (2010). An empirical analysis of offshore service vessel utilization in the US Gulf of Mexico. *International Journal of Energy Sector Management*, *4*, 152–182.
- [21] Leite, R. P. (2012). Maritime transport of deck cargo to petrobras fields in Campos Basin: An empirical analysis, identification and quantification of improvement points. *Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro, Brazil, MSc Thesis*, .
- [22] Mourgaya, M., & Vanderbeck, F. (2007). Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, *183*, 1028–1041.
- [23] Nishimura, E., Imai, A., & Papadimitriou, S. (2001). Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*, *131*, 282–292.
- [24] Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*, 2403–2435.
- [25] Ribeiro, G., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, *39*, 728–735.
- [26] Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*, 455–472.
- [27] Russell, R., & Igo, W. (1979). An assignment routing problem. *Networks*, *9*, 1–17.
- [28] Salazar-Aguilar, M. A., Langevin, A., & Laporte, G. (2012). Synchronized arc routing for snow plowing operations. *Computers & Operations Research*, *39*, 1432–1440.

- [29] Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher, & J.-F. Puget (Eds.), *Principles and Practice of Constraint Programming - CP98* (pp. 417–431). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [30] Shyshou, A., Gribkovskaia, I., Laporte, G., & Fagerholt, K. (2012). A large neighbourhood search heuristic for a periodic supply vessel planning problem arising in offshore oil and gas operations. *INFOR: Information Systems and Operational Research*, 50, 195–204.
- [31] Skonnord, P., & Evans, D. (2015). Offshore rig driller seadrill hopeful to see market turn in 2017. URL: [http://www.rigzone.com/news/article.asp?a\\_id=141009](http://www.rigzone.com/news/article.asp?a_id=141009) accessed March 6, 2018.
- [32] Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35, 254–265.
- [33] Tan, C., & Beasley, J. (1984). A heuristic algorithm for the period vehicle routing problem. *Omega*, 12, 497–504.