A MIP based local search heuristic for a stochastic maritime inventory routing problem *

Agostinho Agra[1], Marielle Christiansen[2], Lars Magnus Hvattum[3], and Filipe Rodrigues[1]

[1] Department of Mathematics and CIDMA, University of Aveiro, Portugal,
{aagra, fmgrodrigues@ua.pt}@ua.pt
[2] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Norway,
mc@iot.ntnu.no
[3] Molde University College, Norway,
hvattum@himolde.no

Abstract. We consider a single product maritime inventory routing problem in which the production and consumption rates are constant over the planning horizon. The problem involves a heterogeneous fleet of ships and multiple production and consumption ports with limited storage capacity. In spite of being one of the most common ways to transport goods, maritime transportation is characterized by high levels of uncertainty. The principal source of uncertainty is the weather conditions, since they have a great influence on sailing times. The travel time between any pair of ports is assumed to be random and to follow a log-logistic distribution. To deal with random sailing times we propose a two-stage stochastic programming problem with recourse. The routing, the order in which the ports are visited, as well as the quantities to load and unload are fixed before the uncertainty is revealed, while the time of the visit to ports and the inventory levels can be adjusted to the scenario. To solve the problem, a MIP based local search heuristic is developed. This new approach is compared with a decomposition algorithm in a computational study.

Keywords: Maritime transportation; Stochastic programming; Uncertainty; Matheuristic.

1 Introduction

We consider a maritime inventory routing problem (MIRP) where a heterogeneous fleet of ships is transporting a single product between ports. There exists one type of ports where the product is produced, and in the other ports the product is consumed. The production and consumption rates are constant over the planning horizon. At all ports, there exists an inventory for storing the product, and lower and upper limits are given for each port. Each port can be visited one or several times during the planning horizon depending on the size of the storage, the production or consumption rate, and the quantity loaded or unloaded at each port visit. The MIRP consists of designing routes and schedules for a fleet of ships in order to minimize the transportation and port costs, and to determine the quantities handled at each port call without exceeding the storage limits. The MIRP is a very important and common problem in maritime shipping and is relevant when the actors involved in a maritime supply

* Published in *Computational Logistics, 7th International Conference*, ICCL 2016, LNCC 9855, 18–34, 2016.

chain have the responsibility for both the transportation of the cargoes and the inventories at the ports. The shipping industry is capital intensive, so a modest improvement in fleet utilization can imply a large increase in profit. Therefore, the ability of ship operators to make good decisions is crucial. However, the MIRPs are very complex to solve due to the high degree of freedom in the routing, scheduling, number of port visits, and the quantity loaded or unloaded at each port visit.

There exists a solid amount of research and resulting publications within MIRPs, and these have formed the basis of several surveys: Papageorgiou et al. [18], Christiansen et al. [10], and Christiansen and Fagerholt [8,9]. In addition, Coelho et al. [12] and Andersson et al. [4] surveyed both land-based and maritime inventory routing problems. Maritime transportation is characterized by high levels of uncertainty, and one of the most common uncertainties is the sailing times that are affected by heavily changing weather conditions. In practice, unpredictable delays may affect the execution of an optimal deterministic plan. In order to compensate for such delays, it is possible for the ships to speed up when necessary. However, in practice it will most often be beneficial to consider the uncertainty explicitly when finding the optimal plan.

Even though maritime transportation is heavily influenced by uncertainty, most of the research reported in the literature on maritime routing and scheduling consider static and deterministic problems. However, some contributions exist, and we describe the ones that are closest to the MIRP with stochastic travel times studied here. For a ship routing and scheduling problem with predefined cargoes, Christiansen and Fagerholt [7] design ship schedules that are less likely to result in ships staying idle at ports during weekends by imposing penalty costs for arrivals at risky times (i.e. close to weekends). The resulting schedule needs to be more robust with respect to delays from bad weather and time in port due to the restricted operating hours in port during weekends. Agra et al. [3] solved a full-load ship routing and scheduling problem with uncertain travel times using robust optimization. Furthermore, Halvorsen-Weare and Fagerholt [14] analysed various heuristic strategies to achieve robust weekly voyages and schedules for off-shore supply vessels working under tough weather conditions. Heuristic strategies for obtaining robust solutions with uncertain sailing times and production rate were also discussed by Halvorsen-Weare et al. [15] for the delivery of liquefied natural gas. For a crude oil transportation and inventory problem, Cheng and Duran [6] developed a decision support system that takes into account uncertainty in sailing time and demand. The problem is formulated as a discrete time Markov decision process and solved by using discrete event simulation and optimal control theory. Rakke et al. [19] and Sherali and Al-Yakoob [20,21] introduced penalty functions for deviating from the customer contracts and the storage limits, respectively, for their MIRPs. Christiansen and Nygreen [11] used soft inventory levels to handle uncertainties in sailing time and time in port, and these levels were transformed into soft time windows for a single product MIRP. Agra et al. [2] were the first to use stochastic programming to model uncertain sailing and port times for a MIRP with several products and inventory management at the consumption ports only. A two-stage stochastic programming model with recourse was developed where the first-stage consists of routing, and loading/unloading decisions, and the second stage consists of scheduling decisions. The model was solved by a decomposition approach similar to an L-shaped algorithm where optimality cuts were added dynamically, and the solution process was embedded within the sample average approximation method.

The objective of this paper is to present a general single product MIRP with stochastic sailing times and a heuristic method to solve the problem. As in the work by Agra et al. [2], we have developed a two-stage stochastic programming model with recourse where the first-stage

consists of routing and loading/unloading decisions, and the second stage consists of scheduling decisions. Although the two problems have several differences (the number of products considered, management inventory at supply ports, and random aspects of uncertainty), this work was also motivated by the stability problems reported for the approach followed by Agra et al. [2]. When the instances become harder, the objective function values obtained by the heuristic approach had large levels of variance. As in previous work we assume the inventory limits can be violated with a penalty. Here we discuss in more detail the impact of the value of such penalties on the stability of the solution procedure, since different penalty values may correspond to different decision maker strategies, and may influence the efficiency of branch and cut based procedures. Low penalty values will be used when backlogged consumption and excess of production are less important than the routing cost, and generate low transportation cost solutions. High penalty values create solutions that are averse to inventory limit violations. Since the fractional solutions obtained by linear relaxations will present, in general, no violation of the inventory limits, the integrality linear gaps tend to be much higher when the penalty values are higher, which deteriorates the performance of branch and cut based procedures. Additionally, in order to circumvent the stability problems, we propose a new heuristic procedure which is based on a local search heuristic that uses the solution from a corresponding deterministic problem as a starting solution.

The remainder of this paper is organized as follows: The mathematical model of the deterministic problem is presented in Section 2, while the stochastic model is presented in Section 3. Section 4 presents the heuristic stochastic solution approaches. Extensive computational results are reported and discussed in Section 5, followed by some concluding remarks in Section 6.

2 Mathematical model for the deterministic problem

In this section we introduce a mathematical formulation for the deterministic problem.

Routing constraints Let V denote the set of ships and N denote the set of ports. Each ship $v \in V$ must depart from its initial position, that can be a point at sea. For each port we consider an ordering of the visits accordingly to the time of the visit.

The ship paths are defined on a network where the nodes are represented by a pair (i, m), where i indicates the port and m indicates the visit number to port i. Direct ship movements (arcs) from node (i, m) to node (j, n) are represented by (i, m, j, n). For ease of notation, if a ship departs from a point at sea, an artificial port is created and a single visit is associated with it.

We define S^A as the set of possible nodes (i, m), S_v^A as the set of nodes that may be visited by ship v, and set S_v^X as the set of all possible movements (i, m, j, n) of ship v.

For the routing we define the following binary variables: x_{imjnv} is 1 if ship v travels from node (i, m) directly to node (j, n), and 0 otherwise; w_{imv} is 1 if ship v visits node (i, m), and 0 otherwise; z_{imv} is equal to 1 if ship v ends its route at node (i, m), and 0 otherwise; y_{im} indicates whether a ship is making the mth visit to port i, (i, m), or not. The parameter $\underline{\mu}_i$ denotes the minimum number of visits at port i and the parameter $\overline{\mu}_i$ denotes an upper

bound on the number of visits at port i.

$$w_{imv} - \sum_{(j,n)\in S_v^A} x_{jnimv} = 0, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{1}$$

$$w_{imv} - \sum_{(j,n)\in S_v^A} x_{imjnv} - z_{imv} = 0, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{2}$$

$$\sum_{v\in V} w_{imv} = y_{im}, \qquad \forall (i,m) \in S^A, \tag{3}$$

$$y_{im} = 1, \qquad \forall (i,m) \in S^A : m \in \{1, \cdots, \underline{\mu}_i\}, \tag{4}$$

$$y_{i(m-1)} - y_{im} \geq 0, \qquad \forall (i,m) \in S^A : \underline{\mu}_i + 1 < m \leq \overline{\mu}_i, \tag{5}$$

$$x_{imjnv} \in \{0,1\}, \qquad \forall v \in V, (i,m,j,n) \in S_v^X, \tag{6}$$

$$w_{imv}, z_{imv} \in \{0,1\}, \qquad \forall v \in V, (i,m) \in S_v^A \tag{7}$$

$$y_{im} \in \{0,1\}, \qquad \forall (i,m) \in S^A. \tag{8}$$

Equations (1) and (2) are the flow conservation constraints, ensuring that a ship arriving at a node also leaves that node or ends its route. Constraints (3) ensure that a ship can visit node (i,m) only if y_{im} is equal to one. Equations (4) fix y_{im} to 1 for the mandatory visits. Constraints (5) state that if port i is visited m times, then it must also have been visited $m-1$ times. Constraints (6)-(8) define the variables as binary.

Loading and unloading constraints Parameter J_i is 1 if port i is a producer; -1 if port i is a consumer. C_v is the capacity of ship v. The minimum and maximum loading and unloading quantities at port i are given by $\underline{Q}_i$ and $\overline{Q}_i$, respectively.

In order to model the loading and unloading constraints, we define the following continuous variables: q_{imv} is the amount loaded or unloaded from ship v at node (i,m); f_{imjnv} denotes the amount that ship v transports from node (i,m) to node (j,n). The loading and unloading constraints are given by:

$$\sum_{(j,n)\in S_v^A} f_{jnimv} + J_i q_{imv} = \sum_{(j,n)\in S_v^A} f_{imjnv}, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{9}$$

$$f_{imjnv} \leq C_v x_{imjnv}, \qquad \forall\ v \in V, (i,m,j,n) \in S_v^X, \tag{10}$$

$$\underline{Q}_i w_{imv} \leq q_{imv} \leq min\{C_v, \overline{Q}_i\} w_{imv}, \qquad \forall v \in V, (i,m) \in S_v^A, \tag{11}$$

$$f_{imjnv} \geq 0, \qquad \forall v \in V, (i,m,j,n) \in S_v^X, \tag{12}$$

$$q_{imv} \geq 0, \qquad \forall v \in V, (i,m) \in S_v^A. \tag{13}$$

Equations (9) are the flow conservation constraints at node (i,m). Constraints (10) require that the ship capacity is obeyed. Constraints (11) impose lower and upper limits on the loading and unloading quantities. Constraints (12)-(13) are the non-negativity constraints.

Time constraints We define the following parameters: T_i^Q is the time required to load/unload one unit of product at port i; T_{ijv} is the travel time between port i and j by ship v. It includes also any set-up time required to operate at port j. T_i^B is the minimum time between two consecutive visits to port i. T is the length of the time horizon, and A_{im} and

B_{im} are the time windows for starting the mth visit to port i. To ease the presentation we also define, for each node (i,m), the following upper bound for the end time of the visit: $T'_{im} = \min\{T, B_{im} + T_i^Q \overline{Q}_i\}$. Given time variables t_{im} that indicate the start time of the mth visit to port i, the time constraints can be written as:

$$t_{im} + \sum_{v \in V} T_i^Q q_{imv} - t_{jn} + \sum_{v \in V | (i,m,j,n) \in S_v^X} \max\{T'_{im} + T_{ijv} - A_{jn}, 0\} x_{imjnv}$$
$$\leq T'_{im} - A_{jn}, \qquad \forall (i,m),(j,n) \in S^A, \tag{14}$$
$$t_{im} - t_{i,m-1} - \sum_{v \in V} T_i^Q q_{i,m-1,v} - T_i^B y_{im} \geq 0, \qquad \forall (i,m) \in S_A : m > 1, \tag{15}$$
$$A_{im} \leq t_{im} \leq B_{im}, \qquad \forall (i,m) \in S^A. \tag{16}$$

Constraints (14) relate the start time associated with node (i,m) to the start time associated with node (j,n) when ship v travels directly from (i,m) to (j,n). Constraints (15) impose a minimum interval between two consecutive visits at port i. Time windows for the start time of visits are given by (16).

Inventory constraints The inventory constraints are considered for each port. They ensure that the stock levels are within the corresponding limits and link the stock levels to the loading or unloading quantities. For each port i, the consumption/production rate, R_i, the minimum $\underline{S}_i$, the maximum $\overline{S}_i$ and the initial S_i^0 stock levels, are given. We define the nonnegative continuous variables s_{im} indicating the stock levels at the start of the mth visit to port i. The inventory constraints are as follows:

$$s_{i1} = S_i^0 + J_i R_i t_{i1}, \qquad \forall i \in N, \tag{17}$$
$$s_{im} = s_{i,m-1} - J_i \sum_{v \in V} q_{i,m-1,v} + J_i R_i (t_{im} - t_{i,m-1}), \qquad \forall (i,m) \in S^A : m > 1, \tag{18}$$
$$s_{im} + \sum_{v \in V} q_{imv} - R_i \sum_{v \in V} T_i^Q q_{imv} \leq \overline{S}_i, \qquad \forall (i,m) \in S^A | J_i = -1, \tag{19}$$
$$s_{im} - \sum_{v \in V} q_{imv} + R_i \sum_{v \in V} T_i^Q q_{imv} \geq \underline{S}_i, \qquad \forall (i,m) \in S^A | J_i = 1, \tag{20}$$
$$s_{i\overline{\mu}_i} + \sum_{v \in V} q_{i,\overline{\mu}_i,v} - R_i (T - t_{i\overline{\mu}_i}) \geq \underline{S}_i, \qquad \forall i \in N | J_i = -1, \tag{21}$$
$$s_{i\overline{\mu}_i} - \sum_{v \in V} q_{i,\overline{\mu}_i,v} + R_i (T - t_{i\overline{\mu}_i}) \leq \overline{S}_i, \qquad \forall i \in N | J_i = 1, \tag{22}$$
$$s_{im} \geq \underline{S}_i, \qquad \forall (i,m) \in S^A | J_i = -1, \tag{23}$$
$$s_{im} \leq \overline{S}_i, \qquad \forall (i,m) \in S^A | J_i = 1. \tag{24}$$

Equations (17) calculate the stock level at the start time of the first visit to a port, and equations (18) relate the stock level at the start time of mth visit to the stock level at the start time of the previous visit. Constraints (19) and (20) ensure that the stock levels are within their limits at the end of each visit. Constrains (21) impose a lower bound on the inventory level at time T for consumption ports, while constrains (22) impose an upper bound on the inventory level at time T for production ports. Constraints (23) and (24) ensure that the stock levels are within their limits at the start of each visit.

Objective function The objective is to minimize the total routing costs, including traveling and operating costs. The traveling cost of ship v from port i to port j is denoted by C^T_{ijv} and it includes the set-up costs. The objective function is defined as follows:

$$Min \qquad C(X) = \sum_{v \in V} \sum_{(i,m,j,n) \in S^X_v} C^T_{ijv} x_{imjnv}. \tag{25}$$

3 Mathematical model for the stochastic problem

In the stochastic approach, the sailing times between ports are assumed to be independent and random, following a known probability distribution (a log-logistic probability distribution which is discussed in Section 5). As in the work by Agra et al. [2], the model introduced here is a recourse model with two levels of decisions. The first-stage decisions are the routing, the port visits sequence, and the load/unload quantities. These decisions must be taken before the scenario is revealed. The corresponding first-stage variables are $x_{imjnv}, z_{imv}, w_{imv}, y_{im}$, and q_{imv}. The adjustable variables are the time of visits and the inventory levels. In the stochastic approach we allow the inventory limits to be violated by including a penalty P_i for each unit of violation of the inventory limits at each port i. In addition to the variables $t_{im}(\xi)$, and $s_{im}(\xi)$ indicating the time and the stock level at node (i,m), when scenario ξ is revealed, new variables $r_{im}(\xi)$ are introduced to denote the inventory limit violation at node (i,m). If i is a consumption port, $r_{im}(\xi)$ denotes the backlogged consumption, that is the amount of demand satisfied with delay. If i is a production port, $r_{im}(\xi)$ denotes the demand in excess to the capacity. We assume the quantity in excess is not lost but a penalty is incurred.

The main goal of the stochastic approach is to find the solution that minimizes the routing cost $C(X)$ plus the expected penalty value for inventory deviation to the limits, $E_\xi(Q(X,\xi))$, where $Q(X,\xi)$ denotes the minimum penalty for the inventory deviations when scenario ξ with a particular sailing times vector is considered and a set of first stage decisions, denoted by X, is fixed. In order to avoid using the theoretical joint probability distribution of the travel times, we follow the common Sample Average Approximation (SAA) method, and replace the true expected penalty value $E_\xi(Q(X,\xi))$ by the mean value of a large random sample $\Omega = \{\xi^1, \ldots, \xi^k\}$ of ξ, obtained by the Monte Carlo method. This larger set of k scenarios is regarded as a benchmark scenario set representing the true distribution [17].

The objective function of the SAA model becomes as follows:

$$Min \qquad C(X) + \frac{1}{\mid \Omega \mid} \sum_{\xi \in \Omega} \sum_{(i,m) \in S^A} P_i r_{im}(\xi). \tag{26}$$

In addition to the routing and loading and unloading constraints (1)-(13), the SAA problem has the following time and inventory constraints.
Time constraints:

$$t_{im}(\xi) + \sum_{v \in V} T^Q_i q_{imv} - t_{jn}(\xi) + \sum_{v \in V \mid (i,m,j,n) \in S^X_v} T^M x_{imjnv},$$
$$\leq T^M, \qquad \forall (i,m),(j,n) \in S^A, \xi \in \Omega, \tag{27}$$

$$t_{im}(\xi) - t_{i,m-1}(\xi) - \sum_{v \in V} T^Q_i q_{i,m-1,v} - T^B_i y_{im} \geq 0, \qquad \forall (i,m) \in S_A : m > 1, \xi \in \Omega, \tag{28}$$

$$A_{im} \leq t_{im}(\xi) \leq B^M_{im}, \qquad \forall (i,m) \in S^A, \xi \in \Omega. \tag{29}$$

The inventory constraints are similar to the constraints for the deterministic problem, but now including the possible violation of the inventory limits. The big constant T^M is now set to $2T$ since the visits to ports can now occur after time period T. Similarly, B^M_{im} is set to $2T$. The inventory constraints are as follows:

$$s_{i1}(\xi) = S^0_i + J_i R_i t_{i1}(\xi) - J_i r_{i1}(\xi), \qquad \forall i \in N, \xi \in \Omega, \tag{30}$$

$$s_{im}(\xi) - J_i r_{i,m-1}(\xi) = s_{i,m-1}(\xi) - J_i r_{im}(\xi) - J_i \sum_{v \in V} q_{i,m-1,v} + J_i R_i (t_{im}(\xi) - t_{i,m-1}(\xi)),$$
$$\forall (i,m) \in S^A : m > 1, \xi \in \Omega, \tag{31}$$

$$s_{im}(\xi) + \sum_{v \in V} q_{imv} - R_i \sum_{v \in V} T^Q_i q_{imv} \leq \overline{S}_i, \qquad \forall (i,m) \in S^A | J_i = -1,\ \xi \in \Omega, \tag{32}$$

$$s_{im}(\xi) - \sum_{v \in V} q_{imv} + R_i \sum_{v \in V} T^Q_i q_{imv} \geq \underline{S}_i, \qquad \forall (i,m) \in S^A | J_i = 1,\ \xi \in \Omega, \tag{33}$$

$$s_{i\overline{\mu}_i}(\xi) + \sum_{v \in V} q_{i,\overline{\mu}_i,v} - R_i (T - t_{i\overline{\mu}_i}(\xi)) + r_{i\overline{\mu}_i}(\xi) \geq \underline{S}_i, \qquad \forall i \in N | J_i = -1, \xi \in \Omega, \tag{34}$$

$$s_{i\overline{\mu}_i}(\xi) - \sum_{v \in V} q_{i,\overline{\mu}_i,v} + R_i (T - t_{i\overline{\mu}_i}(\xi)) - r_{i\overline{\mu}_i}(\xi) \leq \overline{S}_i, \qquad \forall i \in N | J_i = 1, \xi \in \Omega, \tag{35}$$

$$s_{im}(\xi), r_{im}(\xi) \geq 0 \qquad \forall (i,m) \in S^A : m > 1, \xi \in \Omega. \tag{36}$$

For brevity we omit the description of the constraints as their meaning is similar to the meaning of the corresponding constraints for the deterministic problem. The stochastic SAA model is defined by (26) and the constraints (1)-(13), (27)-(36), and will be denoted by SAA-MIRP.

Next we make two important remarks. **Remark 1.** The SAA-MIRP model has relatively complete recourse, since for each feasible solution to the first stage, the inclusion of r variables ensures that the second stage has always a feasible solution. **Remark 2.** When a first stage solution X is known, the second stage variables can easily be obtained by solving k separate linear subproblems.

4 Solution methods

While the deterministic model can be solved to optimality for small size instances, the SAA-MIRP model becomes much harder with the inclusion of the inventory violation variables r, and cannot consistently be solved to optimality for large sample sizes. We consider M separate sets $\Omega_i, i \in \{1, \ldots, M\}$ each one containing $\ell \ll k$ scenarios. The SAA-MIRP model is solved for each set of scenarios Ω_i, (replacing Ω by Ω_i in model SAA-MIRP) giving M candidate solutions. Let us denote by $X^1, \ldots, X^M$, the first stage solutions of those candidate solutions. Then, for each candidate solution the value of the objective function for the large sample $z_k(X^i) = C(X^i) + \frac{1}{k} \sum_{\xi \in \Omega} Q(X^i, \xi)$ is computed and the best solution is determined by $X^* = argmin\{z_k(X^i) : i \in \{1, \ldots, M\}\}$. The average value over all sets of scenarios, $\bar{z}_\ell = \frac{1}{M} \sum_{i=1}^M z^i_\ell$ is a statistical estimate for a lower bound on the optimal value of the true problem and $z_k(X^*)$, is a statistical estimate for an upper bound on the optimal value.

Henceforward we discuss two procedures for solving the SAA-MIRP model for the small sets Ω_i. When employing scenario generation solution procedures it is desirable that no matter which set of scenarios is used, one obtains approximately the same objective function value.

This is referred to as stability requirement conditions [17]. Agra et al. [2] used a decomposition scheme for a stochastic MIRP that was shown to be insufficient to reach stability for hard instances. Here we revisit this procedure and introduce an alternative method.

4.1 Decomposition procedure

A common approach to solve stochastic problems is to decompose the model into a master problem and one subproblem for each scenario, following the idea of the L-shaped algorithm [5]. The master problem consists of the first stage variables and constraints (constraints (1)-(13)), and recourse variables and constraints (27)-(36) defined for a restricted set of scenarios. The subproblems consider fixed first stage decisions, and are solved for each scenario to supply new variables and constraints to the master problem. Since the problem has relatively complete recourse, the resulting subproblems are feasible.

We first solve the master problem including only one scenario to optimality. Then for each disregarded scenario we check whether a penalty for inventory limit violations is incurred when the first stage decision is fixed. If such a scenario is found, we add to the master problem additional variables and constraints enforcing that deviation to be penalized in the objective function. Then the revised master problem is solved again, and the process is repeated until all the recourse constraints are satisfied. Hence, as in the L-shaped method, the master problem initially disregards the recourse penalty, and an improved estimation of the recourse penalty is gradually added to the master problem by solving subproblems and adding the corresponding constraints. A formal description of this process is given below.

Algorithm 1 Decomposition procedure.

```
Consider the master problem with the scenario corresponding to the deterministic problem
Solve the master problem
while There is a scenario ξ ∈ Ω_i leading to an increase of the objective function cost do
    Add constraints (27)-(36) for scenario ξ
    Reoptimize the master problem with the new constraints using a solver for α seconds
end while
```

To check whether there is a scenario $\xi \in \Omega_i$ leading to an increase of the objective function cost, one can use a simple combinatorial algorithm that, for each scenario, determines the earliest arrival time based on the computation of a longest path in an acyclic network [2].

4.2 MIP based local search procedure

In order to circumvent some possible stability problems resulting from the previous procedure, which is based on a truncated branch and cut procedure, we propose a heuristic approach that iteratively searches in the neighborhood of a solution. The procedure starts with the optimal solution from a deterministic model, and ends when no improvement is observed. For the starting solution we either use the deterministic model (1)-(25), with no inventory violations allowed, or the stochastic model containing only one scenario where all travelling times are set to their expected value. To define the neighborhood of a solution, let $\overline{w}$ denote the solution vector of w variables. Following the local branching idea of Fischetti and Lodi [13], we consider as the neighborhood of a solution, the set of solutions that can

differ in at most Δ variables, focusing only on the ship visit variables w_{imv}. This local search can be done by adding the following inequality,

$$\sum_{(i,m)\in S_v^A, v\in V|\overline{w}_{imv}=0} w_{imv} + \sum_{(i,m)\in S_v^A, v\in V|\overline{w}_{imv}=1} (1 - w_{imv}) \leq \Delta. \tag{37}$$

Inequality (37) counts the number of variables w_{imv} that are allowed to flip their value from the value taken in the solution. Note that the routing variables as well as the quantities to load and unload can be changed freely.

In each iteration of the heuristic procedure, the SAA-MIR model restricted with the inclusion of (37) is solved in its extensive form (without the decomposition procedure), since preliminary tests have not shown clear benefits in using the decomposition technique in the restricted model. The procedure is described in Algorithm 2.

Algorithm 2 MIP based Local Search procedure

1: Solve either model (1)-(25), or the SAA-MIRP with a single scenario consisting of expected travel times
2: Set $\overline{w}$ to the optimal value of w
3: **repeat**
4: Add constraint (37) to the model defined for ℓ scenarios
5: Solve the model for α seconds
6: Update the solution $\overline{w}$
7: **until** No improvement in the objective function is observed

5 Computational tests

This section presents some of the computational experiments carried out to test the two solution approaches for a set of instances of a maritime inventory routing problem. The instances are based on real data, and come from the short sea shipping segment with long loading and discharge times relative to the sailing times. These instances result from those presented in [1], with two main differences. One is the computation of the traveling times, which we discuss in detail below, and the other is the production and consumption which we assume here to be constant, where the rates are given by the average of the corresponding values given in the original set of instances. The number of ports and ships of each instance is given in the second column of Table 2. The time horizon is 30 days. Operating and waiting costs are time invariant.

Distribution of travel times and scenario generation

Here we describe the sailing times probability distribution as well as how scenarios are generated. We assume that the sailing times $T_{ijv}(\xi)$ are random and follow a three-parameter log-logistic probability distribution. The cumulative probability distribution can be written as

$$F(T_{ijv}(\xi)) = \frac{1}{1 + (\frac{1}{t})^{\alpha}},$$

where $t = \frac{T_{ijv}(\xi)-\gamma}{\beta}$.

This type of distribution was used in [15] for an LNG (liquefied natural gas) tanker transportation problem, and was motivated by the sailing times calculated for a gas tanker between Rome (Italy) and Bergen (Norway), as reported in [16]. In the three-parameter log-logistic probability distribution, the minimum travel time is equal to γ, and the expected travel time is equal to $E[T_{ijv}(\xi)] = \frac{\beta\pi}{\alpha \sin(\pi/\alpha)} + \gamma$. The three parameters, in [15], were set to $\alpha = 2.24$, $\beta = 9.79$, and $\gamma = 134.47$. In our settings, the deterministic travel time T_{ijv}, given in [1], is set to the expected travel time value, that is, $T_{ijv} = E[T_{ijv}(\xi)]$. In addition, we let $\gamma = 0.9 \times T_{ijv}$, $\alpha = 2.24$ (the same value as in [15], since α is a form parameter), and β is obtained from the equation $T_{ijv} = E[T_{ijv}(\xi)] = \frac{\beta\pi}{\alpha \sin(\pi/\alpha)} + \gamma$. In order to draw a sample, each travel time is randomly generated as follows. First a random number r from $(0, 1]$ is generated. Then the travel time $T_{ijv}(\xi)$ can be found by setting $r = \dfrac{1}{1 + (\frac{1}{t})^{\alpha}}$, which gives $T_{ijv}(\xi) = \gamma + \beta \left(\dfrac{1-r}{r} \right)^{-\frac{1}{\alpha}}$.

Computational results

All tests were run on a computer with an Intel Core i5-2410M processor, having a 2.30GHz CPU and 8GB of RAM, using the optimization software Xpress Optimizer Version 21.01.00 with Xpress Mosel Version 3.2.0.

The number of ports and ships of each instance is given in the second column of Table 1. The following three columns give the size of the deterministic model (1)-(25), and the last three columns give the size for the complete stochastic model SAA-MIRP with $\ell = 25$.

Table 1. Summary statistics for the seven instances.

		Deterministic Model			Stochastic Model		
Inst.	$(\mid N \mid, \mid V \mid)$	# Rows	# Col.	# Int. Var.	# Rows	# Col.	# Int. Var.
A	(4,1)	765	545	273	8413	1713	273
B	(3,2)	767	590	302	5303	1466	302
C	(4,2)	1214	1042	530	8798	2210	530
D	(5,2)	1757	1622	822	13157	3082	822
E	(5,2)	1757	1622	822	13157	3082	822
F	(4,3)	1663	1539	787	9183	2707	787
G	(6,5)	4991	5717	2909	20687	7469	2909

Table 2 gives the optimal values of several instances for the deterministic model. Columns "No violations" give the optimal value (column $C(X)$) and running time in seconds (column *Time*) for the model (1)-(25) with no inventory limit violations allowed. The following columns consider the stochastic model with the expected travel times scenario only. For three different penalty values for inventory limit violations ($P_i = 1 \times \ell$, $P_i = 10 \times \ell$ and $P_i = 100 \times \ell$, where the ℓ is omitted for ease of notation) we provide the routing cost $C(X)$, the value of the inventory violation (columns *Viol*) and the running time (columns *Time*).

Table 2 shows the influence of the penalty on the solution value and instance hardness. The running times are small when $P_i = 1$ and tend to increase with the increase of the penalty. For small penalty values the instances become easier to solve than for the case with hard inventory bounds. When the penalty increases, the integrality gaps also increase (as fractional solutions contain, in general, no violation of the inventory limits) making the instances harder to solve.

Table 2. Instances and the corresponding routing costs and inventory violations for the expected value scenario

	No violations		$P_i = 1$			$P_i = 10$			$P_i = 100$		
Inst.	C(X)	Time	C(X)	Viol	Time	C(X)	Viol	Time	C(X)	Viol	Time
A	130.7	1	6.7	60.0	0	130.7	0.0	0	130.7	0.0	0
B	364.8	6	5.2	235.0	0	364.8	0.0	13	364.8	0.0	19
C	391.5	15	14.7	172.0	0	290.5	3.0	4	324.5	0.5	10
D	347.1	3	55.9	177.0	4	347.1	0.0	42	347.1	0.0	52
E	344.9	343	55.8	184.0	4	344.9	0.0	194	344.9	0.0	181
F	460.9	290	182.0	110.0	3	460.9	0.0	437	460.9	0.0	442
G	645.8	2962	336.3	176.5	17	645.8	0.0	6947	645.8	0.0	16296

For $P_i = 10$, $P_i = 100$ and for the case where violations are not allowed, the solutions coincide for all instances except instance C.

Next we report the results using both procedures following the solution approach described in Section 4 with $M = 10$ sets of scenarios with size $\ell = 25$, and a large sample of size $k = 1000$. In Tables 3, 4, and 5, we present the computational results for the decomposition procedure using branch and cut to solve each master problem with a running time limit of $t = 1$, $t = 2$, and $t = 5$ minutes. After this time limit, if no feasible solution is found, then the running time is extended until the first feasible solution is found. For each table we give the results for the three considered cases of penalties, denoted by $P_i = 1$, $P_i = 10$, and $P_i = 100$. For each penalty we report the following values: the routing cost $C(X)$ of the best solution X^* obtained with the procedure described in Section 4; the average number of violations (columns $Viol$) for solution X^*; the variance between samples $\hat{\sigma}_B^2 = \frac{1}{(M-1)M}\sum_{i=1}^{M}(z_\ell^i - \bar{z}_\ell)^2$; the variance in the larger sample $\hat{\sigma}_L^2 = \frac{1}{(k-1)k}\sum_{\xi\in\Omega}\left(C(X^*) + Q(X^*,\xi) - z_k(X^*)\right)^2$; and the running time, in seconds, of the complete solution procedure. The running time includes solving the M stochastic problems, and for each solution, computing the penalty value for the large set of k samples.

Variances $\hat{\sigma}_B^2$ and $\hat{\sigma}_L^2$ are used to evaluate the stability of the procedure. One can observe that when the penalties increase, the variance for the large sample, $\hat{\sigma}_L^2$, increases as expected. For the variance between samples, $\hat{\sigma}_B^2$, the value also increases when we compare $P_i = 1$ against the other values. Such behavior can be explained by the fact that each master problem is solved by a branch and cut procedure and as explained above, when the penalty increases, the integrality gaps also increase making the instances harder to solve. For those harder instances the branch and cut algorithm acts as a heuristic since the search tree is truncated when the time limit is reached. Thus, the variance tends to increase when we compare those cases where the instances are solved to optimality (some instances with $P_i = 1$) against those cases where the solution procedure acts, in general, as a heuristic (most instances with $P_i = 10$ and $P_i = 100$). However between the cases $P_i = 10$ and $P_i = 100$ there is no obvious trend. There are instances where the decomposition procedure had a better degree of in-sample stability for $P_i = 100$ than for $P_i = 10$. Perhaps as the penalty cost is so high, for some instances the solver identifies the same solution (a solution which is *robust* in relation to inventory bounds violation and minimized the routing cost) for most of the small samples

Ω_i considered. In general, we may state that the decomposition procedure tends to be less stable with the increase of the penalties. There is no clear decrease in the variances when the running time limit is increased.

Table 3. Computational results using the decomposition procedure with a running time limit for each master problem set to 1 minute

	$P_i = 1$					$P_i = 10$					$P_i = 100$				
Inst.	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time
A	130.7	0.0	4	0	105	130.7	0.0	27	0	116	130.7	0.0	27	0	118
B	364.8	1.4	13	0	492	364.8	1.4	64	5	3654	364.8	1.3	57	413	4965
C	263.5	3.0	73	0	151	343.9	2.5	91	0	292	411.5	0.0	36	2	749
D	347.1	2.0	190	0	1627	347.1	2.1	2463	5	3975	347.1	2.2	205	514	3524
E	344.9	3.2	358	0	2277	344.9	5.8	5830	13	5394	260.4	18.4	1945	698	4262
F	501.1	0.1	145	0	327	460.9	0.0	451450	0	5361	460.9	0.0	44413	0	4993
G	433.0	133.9	2136	1	1115	484.8	213.6	71357	151	6962	457.6	212.1	2121500	6	7696

Table 4. Computational results using the decomposition procedure with a running time limit for each master problem set to 2 minutes

	$P_i = 1$					$P_i = 10$					$P_i = 100$				
Inst.	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time
A	130.7	0.0	4	0	102	130.7	0.0	27	0	114	130.7	0.0	27	0	109
B	364.8	1.4	13	0	583	364.8	1.4	1368	5	6143	364.8	1.3	7539	413	8413
C	263.5	3.0	73	0	150	343.9	2.5	91	0	301	391.5	0.0	54	2	928
D	347.1	2.0	190	0	2367	363.2	0.2	792	0	7133	347.1	2.1	314	486	6602
E	344.9	3.2	358	0	2716	363.9	3.2	8839	7	8637	352.9	0.2	1809	40	9032
F	501.1	0.1	145	0	333	460.9	0.0	7025	0	7368	460.9	0.0	4024	0	7824
G	433.0	133.9	2136	1	3260	543.7	154.5	95489	135	10837	442.3	121.9	746118	9670	13003

In Table 6, we report the computational results for the MIP based local search heuristic, starting with a solution obtained by using a single scenario consisting of expected travel times in the SAA-MIRP model. Based on preliminary results, not reported here, we chose $\Delta = 2$. The running time limit is set to 5 minutes, however for most iterations the restricted problem is solved to optimality quickly. In Table 7, we report the corresponding results for the same heuristic but starting with a solution obtained using the model (1)-(25), that is, the model where no deviations to the inventory limits are allowed. We can see that using hard inventory limits for the starting solution leads to a better solution for six instances and worse for two instances.

When comparing the variances with those observed for the decomposition procedure one can observe that the variances between samples are in general lower, meaning that the local search procedure presents a higher degree of in-sample stability than the classical decomposition approach. For the larger sample, both procedures present similar variance values, except for the harder instance (G with $P_i = 100$) where the new heuristic procedure provides better out-of-sample stability. The running times of the local search heuristic are also lower than those for the decomposition procedure.

Table 5. Computational results using the decomposition procedure with a running time limit for each master problem set to 5 minutes

	$P_i = 1$					$P_i = 10$					$P_i = 100$				
Inst.	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time
A	130.7	0.0	4	0	102	130.7	0.0	27	0	116	130.7	0.0	27	0	121
B	364.8	1.4	13	0	602	364.8	1.4	1350	5	11781	364.8	1.3	76	413	17530
C	263.5	3.0	73	0	148	343.9	2.5	91	0	277	391.5	0.0	31	2	1611
D	347.1	2.0	190	0	4120	347.1	2.0	590	5	15130	430.3	1.1	395	127	16889
E	344.9	3.2	358	0	6827	352.9	0.2	1791	1	21592	360.9	3.4	5474	768	25769
F	501.1	0.1	145	0	321	460.9	0.0	36	0	11118	460.9	0.0	11200	0	13218
G	433.0	133.9	2136	1	6421	534.0	57.3	54937	4	20462	517.0	107.3	43724	23544	21564

Table 6. Computational results using the MIP based local search heuristic

	$P_i = 1$					$P_i = 10$					$P_i = 100$				
Inst.	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time
A	130.7	0	4	0	205	130.7	0.0	54	0	210	130.7	0.0	54	0	210
B	260.7	32.7	74	0	406	364.8	1.3	22	4	698	364.8	1.3	44	400	769
C	263.5	3.0	87	0	388	391.5	0.0	11	0	704	411.5	0.0	131	0	944
D	377.2	6.2	28	0	2729	347.1	2.0	1	5	2768	347.1	2.0	48	490	2610
E	405.3	5.6	0	0	3731	344.9	3.1	0	7	3728	344.9	3.1	63	718	3694
F	460.9	0.0	46	0	1484	460.9	0.0	197	0	3438	460.9	0.0	197	0	3118
G	711.9	4.8	239	3	5174	679.5	1.6	602	5	6116	798.9	13.3	731	3246	9686

Table 7. Computational results using the MIP based local search heuristic with the starting solution obtained for the deterministic model with hard inventory constraints

	$P_i = 1$					$P_i = 10$					$P_i = 100$				
Inst.	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time	$C(X)$	Viol	$\hat{\sigma}_B^2$	$\hat{\sigma}_L^2$	Time
A	130.7	0	4	0	72	130.7	0.0	54	0	89	130.7	0.0	54	0	96
B	345.2	3.8	24	0	383	364.8	1.3	22	4	722	364.8	1.3	44	400	719
C	263.5	3.0	87	0	266	354.5	0.5	38	0	533	391.5	0.0	57	0	630
D	347.2	2.0	28	0	1097	347.1	2.0	1	5	2992	347.1	2.0	48	490	3675
E	344.9	3.1	0	0	1214	344.9	3.1	0	7	3227	344.9	3.1	63	718	5822
F	460.9	0.0	46	0	941	460.9	0.0	460	0	1670	460.9	0.0	197	0	1388
G	654.9	1.8	138	0	4592	679.5	1.6	994	5	8130	912.5	0.6	402	112	10332

Finally, in Table 8 we present the overall cost $z_k(X)$ for the best solution obtained with the two solution procedures. Columns *Decomp.* give the cost value for the decomposition procedure using a time limit of 5 minutes, and columns *MIPLS* give the corresponding value for the MIP based local search heuristic using the starting solution with no inventory deviations. The best result from the two approaches is highlighted in bold.

Table 8. Cost $z_k(X)$ of the best solution obtained with each one of the solution procedures

	$P_i = 1$		$P_i = 10$		$P_i = 100$	
Inst.	*Decomp.*	*MIPLS*	*Decomp.*	*MIPLS*	*Decomp.*	*MIPLS*
A	130.7	130.7	130.7	130.7	130.7	130.7
B	441.0	441.0	725.6	**696.0**	3707.2	**3632.3**
C	338.5	338.5	968.9	**479.5**	454.1	**391.5**
D	397.7	397.7	857.0	**853.9**	**3261.5**	5414.2
E	422.0	422.0	**405.1**	1115.2	8804.7	**8047.7**
F	502.9	**460.9**	460.9	460.9	460.9	460.9
G	3780.8	**692.4**	14861.1	**1085.8**	268712.0	**2359.6**

We can see that the new MIP based local search procedure is better than the decomposition procedure in ten instances and worse in two. The decomposition procedure performs well when instances can be solved to optimality. Overall, we may conclude that the local search heuristic is more attractive than the decomposition procedure based on the branch and cut when the instances are not solved to optimality, since the local search heuristic is faster, presents better levels of in-sample stability for almost all instances and better levels of out-of-sample stability for the hardest instance, and provides good quality solutions. On the other hand, for the instances that can be solved to optimality, the decomposition procedure is the best option.

6 Conclusions

We consider a maritime inventory routing problem where the travel times are stochastic. The problem is modeled as a two-stage stochastic programming problem with recourse, where violations of inventory limits are penalized. A decomposition procedure that solves the master problem using a commercial solver and a MIP based local search algorithm, are proposed. For several instances the master problem is not solved to optimality within reasonable running times. Hence both procedures can be regarded as heuristics. The two procedures are tested for stability using different values for the penalties. A computational study based on a small set of benchmark instances shows that when the penalties are low, the instances are easier to solve by exact methods, and the decomposition procedure can be used efficiently. On the other hand, when penalties are high, the integrality gaps tend to increase making the decomposition procedure, that uses the branch and cut to solve the master problem, less stable than the MIP based local search heuristic. Additionally, the new proposed heuristic is in general faster than the decomposition procedure.

Aknowledgements

The work of the first author was funded by FCT (Fundação para a Ciência e a Tecnologia) and CIDMA (Centro de Investigação e Desenvolvimento em Matemática e Aplicações) within project UID/MAT/04106/2013. The work or the fourth author was funded by FCT under Grant PD/BD/114185/2016.

References

1. Agra, A., Andersson, H., Christiansen, M., Wolsey, L.: A maritime inventory routing problem: Discrete time formulations and valid inequalities. Networks 64, 297–314 (2013)
2. Agra, A., Christiansen, M., Delgado, A., Hvattum, L.M.: A maritime inventory routing problem with stochastic sailing and port times. Computers and Operations Research 61, 18–30 (2015)
3. Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L.M., Poss, M., Requejo, C.: The robust vehicle routing problem with time windows. Computers and Operations Research 40, 856–866 (2013)
4. Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A.: Industrial aspects and literature survey: Combined inventory management and routing. Computers and Operations Research 37, 1515–1536 (2010)
5. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming, Springer Series in Operations Research and Financial Engineering, 2nd edition, New York (2011)
6. Cheng, L., Duran, M.A.: Logistics for world-wide crude oil transportation using discrete event simulation and optimal control. Computers and Chemical Engineering 28, 897–911 (2004)
7. Christiansen, M., Fagerholt K.: Robust Ship Scheduling with Multiple Time Windows. Naval Research and Logistics 49, 611–625 (2002)
8. Christiansen, M., Fagerholt, K.: Maritime inventory routing problems. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 1947–1955. Springer (2009)
9. Christiansen, M., Fagerholt, K.: Ship routing and scheduling in industrial and tramp shipping. In: Toth, P., Vigo, D. (eds.) Vehicle Routing: Problems, Methods, and Applications, pp. 381–408. MOS-SIAM Series on Optimization, Philadelphia (2014)
10. Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D.: Ship routing and scheduling in the new millennium. European Journal of Operational Research 228, 467–483 (2013)
11. Christiansen, M., Nygreen, B.: Robust inventory ship routing by column generation. In: Desrosiers, G., Solomon, M.M. (eds.) Column Generation, pp. 197–224. Springer-Verlag, New York (2005)
12. Coelho, L.C., Cordeau, J.-F., Laporte, G.: Thirty Years of Inventory Routing. Transportation Science 48, 1–19 (2014)
13. Fischetti, M., Lodi., A.: Local Branching. Mathematical Programming Ser. B 98, 23–47 (2003)
14. Halvorsen-Weare, E.E., Fagerholt, K.: Robust supply vessel planning. In: Pahl, J., Reiners, T., Voß, S. (eds.) Network Optimization, LNCS vol. 6701, pp. 559–573. Springer, Berlin (2011)
15. Halvorsen-Weare, E.E., Fagerholt, K., Rönnqvist, M.: Vessel routing and scheduling under uncertainty in the liquefied natural gas business. Computers and Industrial Engineering 64, 290–301 (2013)
16. Kauczynkski, W.: Study of the reliability of the ship transportation. In: Proceeding of the International Conference on Ship and Marine Research, 15 pages. (1994)
17. Kaut, M., Wallace, S.: Evaluation of scenario-generation methods for stochastic programming, Pacific Journal of Optimization 3, 257–271 (2007)
18. Papageorgiou, D.J., Nemhauser, G.L., Sokol, J., Cheo, 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 234, 350–366 (2014)

19. Rakke, J.G., Stålhane, M., Moe, C.R., Andersson, H., Christiansen, M., Fagerholt, K., Norstad, I.: A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. Transportation Research, Part C 19, 896–911 (2011)
20. Sherali, H.D., Al-Yakoob, S.M.: Determining an optimal fleet mix and schedules: Part I - Single source and destination. In: Karlof, J.K. (ed.) Integer Programming Theory and Practice, pp. 137–166. CRC Press, Boca-Raton (2006)
21. Sherali, H.D., Al-Yakoob, S.M., Determining an optimal fleet mix and schedules: Part II - Multiple sources and destinations, and the option of leasing transshipment depots. In: Karlof, J.K. (ed.) Integer Programming Theory and Practice, pp. 167–194. CRC Press, Boca-Raton (2006)