A Maritime Inventory Routing Problem with Stochastic Sailing and Port Times

Agra, Agostinho
aagra@ua.pt

Christiansen, Marielle
mc@iot.ntnu.no

Delgado, Alexandrino
alexandrino.delgado@docente.unicv.edu.cv

Hvattum, Lars Magnus
lars.m.hvattum@iot.ntnu.no

30 May 2013

Abstract

We consider a stochastic short sea shipping problem where a company is responsible for both the distribution of oil products between islands and the inventory management of those products at unloading ports. Ship routing and scheduling is associated to uncertainty in weather conditions and unpredictable waiting times in ports, and in this work, both sailing times and port times are considered to be stochastic parameters.

A two-stage stochastic programming model with recourse is presented where the first-stage consists of routing, loading and unloading decisions, and the second stage consists of scheduling decisions. The model is solved using a decomposition approach similar to an L-shaped algorithm where optimality cuts are added dynamically, and this solution process is embedded within the sample average approximation method. A computational study based on ten real-world instances is presented.

Keywords: Stochastic programming; Maritime transportation; Uncertainty; L-shaped method; Sample average approximation; Travel time; Service time.

1 Introduction

Maritime transportation is characterized by high levels of uncertainty. In practice, operational plans are often adjusted due to factors such as changing weather conditions, ports congestions, or mechanical problems at port. A plan that minimizes the transportation and port costs based on expected sailing and port times may not necessarily be good, as it does not account for consequences resulting from delays. Hence, in most practical situations it will be beneficial to consider the possibility of delays when trying to minimize costs.

In this paper we consider a maritime inventory routing problem occurring at the archipelago of Cape Verde. A deterministic variant of this problem was solved to optimality in [1] for short time horizons. Heuristics for the same problem with time horizons up to 6 months were developed in [2]. The deterministic methods assume known and fixed sailing times, but the planner needs to face the uncertainty associated with the ships sailing between ports. This may somehow be circumvented by the inclusion of safety stocks at inventories or by artificially increasing the sailing times to compensate for delays. However, in this paper we consider explicitly uncertainty in both sailing times between ports and waiting times at ports, over a short time horizon. Bad weather can lead to both longer sailing and port times. The ports are used by several independent shipping companies, and limited coordination between the various operators can result in heavy port congestion. This may come from limited capacities in the inner port area, at berths, and of pipes and other important equipment for performing the (un-)loading operations. In addition, delays may occur due to mechanical problems at port. By taking this into account, good distribution plans can be found that explicitly takes into account the real possibility of violating inventory limits at production or consumption ports.

This paper describes a stochastic programming model with recourse where the routes and the quantities to load and unload must be fixed *a priori*, that is, before actual values of the uncertain parameters are revealed, while the schedule of the loading and unloading operations can be adjusted according to the observed sailing and port times.

The solution method combines the use of the sample average approximation method with a decomposition procedure resembling an L-shaped method [5, 11]. For a given set of scenarios, the corresponding two-stage model is solved to obtain a candidate solution. This is repeated for several different sets of scenarios to obtain several candidate solutions. To choose the best solution, these candidate solutions are evaluated for a larger and independent set of scenarios. To solve the two-stage model for a given set of scenarios, the problem is decomposed into a master problem and one subproblem for each scenario, where the second-stage decisions are considered in the subproblems. We show that feasibility is always guaranteed for the solution obtained in the first stage. Then we show how to derive optimality cuts from the subproblems that are added dynamically to the master problem.

The remainder of this paper is organized as follows. In Section 2 we describe the real problem and review some relevant literature. Then, in Section 3 we present a scenario based mathematical formulation for the problem. The solution approach based on decomposing the problem is discussed in Section 4. In Section 5 we describe how the stochastic sailing and port times have been modeled, and how scenarios have been generated. Section 6 contains computational results for ten real-world instances, and in Section 7 we present the main conclusions of this work.

2 Problem description and literature review

In Cape Verde, fuel oil products are imported and delivered to specific islands and stored in large supply storage tanks. From these islands, fuel oil products are distributed among all inhabited islands using a small heterogeneous fleet of ships. Products are stored in separate consumption storage tanks with limited capacity. Some ports have both supply tanks for some products and consumption tanks for other products. As the capacities of the supply tanks are very large compared to the total consumption over the planning horizon, the inventory aspects for these tanks can be ignored. Not all islands consume all products. Consumption rates are assumed to be constant over the time horizon. Each port can receive at most one ship at a time, and in some ports there exists a minimum time interval between the departure of one ship and the arrival of the next ship.

Each ship has a specified capacity, fixed speed, and cost structure. The cargo hold of each ship is separated into several cargo tanks. The products cannot be mixed, so we assume that the ships have dedicated tanks for the particular products. The ships are either sailing, waiting outside a port or operating. Here, operating is the common term for loading and unloading.

At port, we consider set-up times for the coupling and decoupling of pipes and operation times which depend on the amount loaded or unloaded. Minimum and maximum unloading quantities can be derived. The maximum unloading quantity is imposed by the inventory capacity at the consumption port and by the ship cargo tank capacity.

The driving force in the problem is the need for fuel oil products in the consumption storage tanks. If the demand is not satisfied, the backlogged demand will be penalized by a cost.

The traveling times depend upon the weather conditions and are considered stochastic. The uncertain time parameter at port is manly related to the time from arrival to start of operation. Hence, a specified waiting time before start of service is defined as stochastic, while the operation times are deterministic.

The inter-island distribution plan consists of routes and schedules for the fleet of ships, and describes the number of visits to each port and the quantity of each product to be loaded or unloaded at each port visit. This plan must satisfy the capacities of the ships and consumption inventories while minimizing the sailing and port costs as well as the expected penalty costs of backlogged demand. There is great flexibility in the route pattern of a ship, such that a ship may visit several loading ports as well as unloading ports in succession and the quantities loaded or unloaded are variable as well as the number of visits at each port. The problem described here will be referred to as a *stochastic maritime inventory routing problem* (SMIRP), and a scenario based stochastic programming model for the problem is given in Section 3.

The amount of literature on maritime transportation optimization has increased steadily over the last decades, as evidenced through the recent survey in [7]. Despite being a transportation mode that is heavily influenced by uncertainty, most of the literature on maritime routing and scheduling involves solving static and deterministic problem variants. However, some contributions exist, and we describe some that are considering problems close to the stochastic maritime inventory routing problem of this paper.

An inventory routing problem with uncertain demands and sailing times was solved heuristically by Cheng and Duran [6]. Rakke et al. [17] and Sherali and Al-Yakoob [18, 19] introduce penalty functions for deviating from the customer contracts and the inventory limits, respectively. Christiansen and Nygreen [8] introduce soft inventory levels to handle uncertainties in sailing and port times, and these levels are transformed into soft time windows.

Agra et al. [3] solved a full-load ship routing and scheduling problem with uncertain travel times using robust optimization. Weather conditions affect both sailing speeds and the loading and unloading operations for supply vessels servicing offshore installations, and various heuristic strategies to achieve robust weekly voyages and schedules were analyzed by Halvorsen-Weare and Fagerholt [9]. Heuristic strategies for obtaining robust solutions with uncertain sailing times was also discussed by Halvorsen-Weare et al. [10] for the delivery of liquefied natural gas. None of the aforementioned research has used stochastic programming to model uncertain sailing and port times.

A stochastic model for a particular version of the *vehicle routing problem* (VRP) with stochastic travel times was presented by Lambert et al. [14], and a heuristic solution method was proposed. Considering a VRP with stochastic travel times and service times, Laporte et al. [15] presented a chance constrained formulation as well as two recourse formulations. The recourse problem was solved to optimality for up to 20 nodes and 5 scenarios using an integer L-shaped method. The VRP with stochastic travel and service times was also studied by Kenyon and Morton [13], considering stochastic programming models that minimized the expected completion time or maximized the probability of completing the routes within a given deadline. An integer L-shaped algorithm was used by Teng et al. [20] to solve a time-constrained *traveling salesman problem* with stochastic travel and service times with up to 35 nodes. Although these papers present stochastic programming models for routing problems with uncertain travel times and service times, they do not consider heterogeneous fleets, a variable number of visits, nor inventory constraints.

3 Mathematical Model

In this section we introduce a two-stage stochastic programming model with recourse for the SMIRP problem. The routes and the quantities to load and unload are determined in the first stage. However, the schedule of the loading and unloading operations can be adjusted in the second stage. Thus, also the inventory level variables are allowed to change according to the realization of the stochastic parameters. In the following we first describe the variables and constraints related to the first stage (Section 3.1), and then the variables and constraints related to the second stage (Section 3.2).

3.1 First stage

First we model the routing and the loading and unloading constraints.

Routing constraints:

Let V denote the set of ships. Each ship $v \in V$ must depart from its initial position in the beginning of the planning horizon. For each port we consider an ordering of the visits accordingly to the time of visit. The ship paths are defined on a network where the nodes are represented by a pair (i, m), where i is the port and m is the m^{th} visit to port i. A direct ship movement (arc) from port arrival (i, m) to port arrival (j, n) is represented by (i, m, j, n).

We define S^A as the set of possible port arrivals (i, m), S_v^A as the set of port arrivals that may be visited by ship v, S^X as the set of all possible ship movements (i, m, j, n), and set S_v^X as the set of all possible movements of ship v. The set of ships that can visit port i is denoted V_i.

For the routing we define the following binary variables: x_{imjnv} that is 1 if ship v sails from port arrival (i, m) directly to port arrival (j, n), and 0 otherwise; x_{imv}^O that indicates whether ship v sails directly from its initial position to port arrival (i, m) or not; w_{imv} is 1 if ship v visits port i at arrival (i, m), and 0 otherwise; z_{imv} is equal to 1 if ship v ends its route at port arrival (i, m), and 0 otherwise; z_v^O is equal to 1 if ship v is not used and 0 otherwise; y_{im} indicates whether a ship is visiting port arrival (i,m) or not.

$$\sum_{(i,m)\in S_v^A} x_{imv}^O + z_v^O = 1, \qquad v \in V, \tag{1}$$

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

$$w_{imv} - \sum_{(j,n)\in S_v^A} x_{imjnv} - z_{imv} = 0, \qquad v \in V, (i,m) \in S_v^A, \tag{3}$$

$$\sum_{v\in V_i} w_{imv} = y_{im}, \qquad (i,m) \in S^A, \tag{4}$$

$$y_{i(m-1)} - y_{im} \geq 0, \qquad (i,m) \in S^A : m > 1, \tag{5}$$

$$x_{imv}^O, w_{imv}, z_{imv} \in \{0,1\}, \qquad v \in V, (i,m) \in S_v^A, \tag{6}$$

$$x_{imjnv} \in \{0,1\}, \qquad v \in V, (i,m,j,n) \in S_v^X, \tag{7}$$

$$z_v^O \in \{0,1\}, \qquad v \in V, \tag{8}$$

$$y_{im} \in \{0,1\}, \qquad (i,m) \in S^A. \tag{9}$$

Equations (1) ensure that each ship either departs from its initial position and sails towards another port or the ship is not used. Equations (2) and (3) are the arc flow conservation constraints, ensuring that a ship arriving at a port also leaves that port or ends its route. Constraints (4) ensure that one ship only visits port (i,m) if y_{im} is equal to one. Constraints (5) state that if port i is visited m times, then it must also have been visited $m-1$ times. Constraints (6) - (9) define the variables as binary.

Loading and unloading constraints

Let K represent the set of products and K_v represent the set of products that ship v can transport. Not all ports consume all products. Parameter J_{ik} is 1 if port i is a supplier of product k; -1 if port i is a consumer of product k, and 0 if i is neither a consumer nor a supplier of product k. The quantity of product k on board of ship v at the beginning of the planning horizon is given by Q_{vk}^O and C_{vk} is the capacity of the compartment of ship v dedicated for product k. The minimum and the maximum discharge quantities of product k at port i are given by $\underline{Q}_{ik}$ and $\overline{Q}_{ik}$, respectively. Parameter T is the length of the time horizon.

To model the loading and unloading constraints, we define the following binary variables: o_{imvk} is equal to 1 if product k is loaded onto or unloaded from ship v at port visit (i,m), and 0 otherwise. In addition, we define the following continuous variables: q_{imvk} is the amount of product k loaded onto or unloaded from ship v at port visit (i,m); f_{imjnvk} denotes the amount of product k that ship v transports from port visit (i,m) to port visit (j,n), and f_{imvk}^O gives the amount of product k that ship v transports from its initial position to port visit (i,m).

The loading and unloading constraints are given by:

$$f^O_{jnvk} + \sum_{(i,m)\in S^A_v} f_{imjnvk} + J_{jk}q_{jnvk} = \sum_{(i,m)\in S^A_v} f_{jnimvk}, \qquad v \in V, (j,n) \in S^A_v, k \in K_v, \tag{10}$$

$$f^O_{imvk} = Q^O_{vk} x^O_{imv}, \qquad v \in V, (i,m) \in S^A_v, k \in K_v, \tag{11}$$

$$f_{imjnvk} \leq C_{vk} x_{imjnv}, \qquad v \in V, (i,m,j,n) \in S^X_v, k \in K_v, \tag{12}$$

$$0 \leq q_{imvk} \leq C_{vk} o_{imvk}, \qquad v \in V, (i,m) \in S^A_v, k \in K_v : J_{ik} = 1, \tag{13}$$

$$\underline{Q}_{ik} o_{imvk} \leq q_{imvk} \leq \overline{Q}_{ik} o_{imvk}, \qquad v \in V, (i,m) \in S^A_v, k \in K_v : J_{ik} = -1, \tag{14}$$

$$\sum_{k\in K_v} o_{imvk} \geq w_{imv}, \qquad v \in V, (i,m) \in S^A_v, \tag{15}$$

$$\sum_{(i,m)\in S_{A_v}} \sum_{v\in V} \sum_{k\in K_v : J_{ik}=-1} q_{imvk} \geq \sum_{i\in N} \sum_{k\in K : J_{ik}=-1} R_{ik}T, \tag{16}$$

$$o_{imvk} \leq w_{imv}, \qquad v \in V, (i,m) \in S^A_v, k \in K_v, \tag{17}$$

$$f_{imjnvk} \geq 0, \qquad v \in V, (i,m,j,n) \in S^A_v, k \in K_v, \tag{18}$$

$$f^O_{imvk}, q_{imvk} \geq 0, \qquad v \in V, (i,m) \in S^A_v, k \in K_v, \tag{19}$$

$$o_{imvk} \in \{0,1\}, \qquad v \in V, (i,m) \in S^A_v, k \in K_v. \tag{20}$$

Equations (10) are the load flow conservation constraints. Equations (11) determine the quantity on board when ship v sails from its initial port position to port arrival (i,m). Constraints (12) guarantee that the ships' tank capacities are not exceeded. Constraints (13) impose an upper bound on the quantity loaded at the supply ports. Constraints (14) impose lower and upper limits on the unloaded quantities. Constraints (15) ensure that if ship v visits port arrival (i,m), then at least one product must be (un)loaded. Constraints (16) ensure that the sum of delivered goods should not be less than the sum of the consumption over the entire horizon T. Constraints (17) ensure that if ship v (un)loads one product at visit (i,m), then w_{imv} must be one. Constraints (18)-(20) are the non-negativity and integrality requirements.

3.2 Second stage

Now we present the second stage model where the variables can be adjusted to the scenario. The set of scenarios Ω will be indexed by c.

Time constraints

To keep track of the inventory level it is necessary to determine the start and the end times at each port arrival. We define the following parameters: T^Q_{ik} is the time required to load/unload one unit of product k at port i; T^S_{ik} is the set up time required to operate product k at port i. T_{ijvc} is the sailing time between port i and j by ship v for scenario c; T^O_{ivc} indicates the sailing time required by ship v to travel from its initial port position to port i for scenario c; T^B_i is the minimum interval between the departure of one ship and the next arrival at port i; T^W_{imc} is the waiting time at port arrival (i,m) for scenario c. The parameter μ_i denotes the maximum number of visits at port i. For each scenario c we define the start time t_{imc} and the end time t^E_{imc} variables for port arrival (i,m). Variables t^+_{ic} give the remaining time from the end of the last visit at port i until time T for scenario c, when this visit occurs before time T.

Assuming that a ship travels from (i,m) to (j,n) under scenario c and loads product k using vessel v, Figure 1 shows the parameters involved when calculating the time variables for node (j,n).

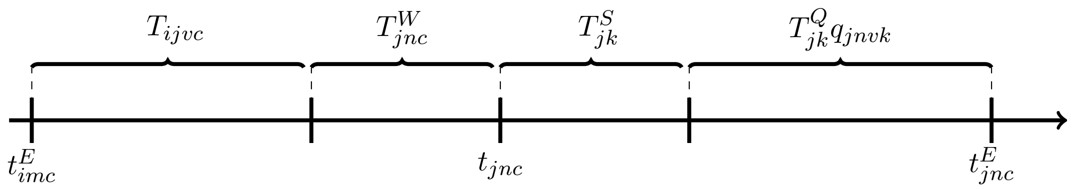

Figure 1: Illustration of the parameters involved when calculating start and end times for node (j,n).

The set of time constraints is as follow:

$$t^E_{imc} \geq t_{imc} + \sum_{v\in V}\sum_{k\in K_v} T^S_{ik} o_{imvk} + \sum_{v\in V}\sum_{k\in K_v} T^Q_{ik} q_{imvk}, \qquad (i,m)\in S^A, c\in\Omega, \tag{21}$$

$$t_{imc} - t^E_{i(m-1)c} - T^B_i y_{im} \geq 0, \qquad (i,m)\in S^A : m>1, c\in\Omega, \tag{22}$$

$$t^E_{imc} + \sum_{v\in V_i\cap V_j} T_{ijvc} x_{imjnv} + T^W_{jnc} - t_{jnc} \leq M(1 - \sum_{v\in V_i\cap V_j} x_{imjnv}), (i,m,j,n)\in S^X, c\in\Omega, \tag{23}$$

$$\sum_{v\in V} T^O_{ivc} x^O_{imv} + T^W_{imc} \leq t_{imc}, \qquad (i,m)\in S^A, c\in\Omega, \tag{24}$$

$$t^+_{ic} \geq T - t^E_{i\overline{\mu}_i c}, \qquad i\in N, c\in\Omega, \tag{25}$$

$$t_{imc}, t^E_{imc} \geq 0, \qquad (i,m)\in S^A, c\in\Omega, \tag{26}$$

$$t^+_{ic} \geq 0, \qquad i\in N, c\in\Omega. \tag{27}$$

Constraints (21) define the end time of service at port visit (i,m). Constraints (22) impose a minimum interval between two consecutive visits at port i. Constraints (23) relate the end time of port visit (i,m) to the start time of port visit (j,n) when ship v sails directly from port (i,m) to (j,n). The big-M constant, denoted by M was set to $2T$, since the start time of a visit can occur after time T. These constraints are a stronger version of the usual family of constraints $t^E_{imc} + T_{ijvc} + T^W_{jnc} - t_{jnc} \leq M(1 - x_{imjnv})$ defined for each $v\in V$. Constraints (24) ensure that if ship v travels from its initial position directly to port visit (i,m), then the start time is at least the sailing time between the two positions plus the waiting time at port visit (i,m). Constraints (25) together with (27) determine the time gap between the last visit to port i and time T. The continuous time variables are declared as non-negative in (26) and (27).

Inventory constraints

The inventory constraints are considered for each unloading port i ($J_{ik} = -1$). They ensure that the inventory levels are kept within the corresponding bounds, and link the inventory levels to the unloaded quantities.

For each consumption port i, and for each product k, the demand rate, R_{ik}, the minimum $\underline{S}_{ik}$, the maximum $\overline{S}_{ik}$, and the initial S^O_{ik} inventory levels are given.

We define the nonnegative continuous variables s_{imkc} and s^E_{imkc} indicating the inventory levels at the start and at the end of port visit (i,m) for scenario c, respectively; r_{imkc} and r^E_{imkc} indicate the backlog of product k at the start and at the end of port visit (i, m) for scenario c, respectively. The inventory constraints are as follow:

$$s_{i1kc} = S^O_{ik} - R_{ik}t_{i1c} + r_{i1kc}, \qquad i \in N, k \in K : J_{ik} = -1, c \in \Omega, \tag{28}$$

$$s^E_{imkc} + r_{imkc} = s_{imkc} + r^E_{imkc} + \sum_{v \in V} q_{imvk} - R_{ik}(t^E_{imc} - t_{imc}), \qquad (i,m) \in S^A, \\ k \in K : J_{ik} = -1, c \in \Omega, \tag{29}$$

$$s_{imkc} + r^E_{i(m-1)kc} = s^E_{i(m-1)kc} + r_{imkc} - R_{ik}(t_{imc} - t^E_{i(m-1)c}), \qquad (i,m) \in S^A : m > 1, \\ k \in K : J_{ik} = -1, c \in \Omega, \tag{30}$$

$$s_{imkc}, s^E_{imkc} \leq \overline{S}_{ik}, \qquad (i,m) \in S^A, k \in K : J_{ik} = -1, c \in \Omega, \tag{31}$$

$$s^E_{i\mu_i kc} - R_{ik}t^+_{ic} \geq \underline{S}_{ik}, \qquad i \in N, k \in K : J_{ik} = -1, c \in \Omega, \tag{32}$$

$$s_{imkc}, s^E_{imkc}, r_{imkc}, r^E_{imkc} \geq 0, \qquad (i,m) \in S^A, k \in K : J_{ik} = -1, c \in \Omega. \tag{33}$$

Equations (28) calculate the inventory level of each product at the first visit. Equations (29) calculate the inventory level of each product when the service ends at port visit (i, m). Equations (30) relate the inventory level at the start of port visit (i, m) to the inventory level at the end of port visit $(i, m - 1)$. Constraints (31) ensure that the capacities of the depots are not exceeded. Constraints (32) impose a lower bound on the inventory level at time T, or at the end of the last visit, for each product. When the last visit occurs before T, the inventory level at the last visit needs to be reduced by the consumption until time T. The quantity below this lower bound is penalized as backlogged demand. Finally, non-negativity requirements (33) are imposed on the inventory and backlog variables.

3.3 Objective function

The objective is to minimize the sailing, setup and operating costs plus the penalty for backlogged demand. The sailing cost of ship v from port i to port j is denoted by C^T_{ijv}, while C^{TO}_{oiv} represents the sailing cost of ship v from its initial port position to port i. The operating cost of product k at port i is denoted by C^O_{ik}. The penalty cost for backlogging of product k at port i is denoted C^P_{ik}. The objective function is as follow:

$$z = \min \sum_{v \in V} \sum_{(i,m,j,n) \in S^X_v} C^T_{ijv} x_{imjnv} + \sum_{v \in V} \sum_{(i,m) \in S^A_v} C^{TO}_{oiv} x^O_{imv} + \\ \sum_{v \in V} \sum_{(i,m) \in S^A_v} \sum_{k \in K_v} C^O_{ik} o_{imvk} + \sum_{c \in \Omega} \frac{1}{|\Omega|} \left(\sum_{(i,m) \in S^A} \sum_{k \in K_v} C^P_{ik}(r_{imkc} + r^E_{imkc}) \right). \tag{34}$$

We penalize backlogged demand for each port visit. For the last visit, or time T if the last visit is earlier, we penalize the difference between the lower bound of the inventory and the actual inventory level in addition to any backlog.

4 Solution approach

Since the complete model is too large to be solved efficiently, it is decomposed into a master problem and one subproblem for each scenario, following the idea of the L-shaped algorithm [5]. Let the problem (1) - (34) be re-written as:

$$z = \min\ C(X) + \sum_{c\in\Omega} \frac{1}{|\Omega|} H(X,c)$$
$$s.t.\ (1) - (20)$$

where

$$C(X) = \sum_{v\in V} \sum_{(i,m,j,n)\in S_v^X} C_{ijv}^T x_{imjnv} + \sum_{v\in V} \sum_{(i,m)\in S_v^A} C_{oiv}^{TO} x_{imv}^O$$
$$+ \sum_{v\in V} \sum_{(i,m)\in S_v^A} \sum_{k\in K_v} C_{ik}^O o_{imvk}$$

and

$$H(X,c) = \min \sum_{(i,m)\in S^A} \sum_{k\in K: J_{ik}=-1} C_{ik}^P (r_{imkc} + r_{imkc}^E)$$
$$s.t.\ (21) - (33), \text{with } \Omega = \{c\}.$$

The master problem consists of the first stage, but with iteratively added variables and constraints to reflect the recourse costs. The subproblems consider fixed first stage decisions, and are solved for each scenario to supply optimality cuts to the master problem.

The problem (1) - (34) has relatively complete recourse, since feasibility in the second stage is guaranteed if the inventory levels do not exceed the capacities of the inventories. Hence, for each feasible solution to the first stage, the second stage has always a feasible solution (it suffices to delay the unloading when necessary). The details for solving the problem are given in Section 4.2. To solve problems with a large number of scenarios, the sample average approximation method is used as described in Section 4.1.

4.1 Sample average approximation method

To solve the SMIRP with many scenarios, we apply the sample average approximation method [21]. First we consider M separate sets of scenarios. Each set of scenarios, $i \in \{1, \dots M\}$ contains a small number of m scenarios, $\{c^{i1}, \dots, c^{im}\}$. The model (1) - (34) is solved for each set of scenarios i using a decomposition approach. Let X^i denote the obtained first stage solution. The M candidate solutions $X^1, \dots, X^M$, are then compared using a different, and much larger, set of n scenarios $\{\hat{c}^1, \dots, \hat{c}^n\}$. The best solution is given by $X^* = argmin\{z_n(X^i) : i \in \{1, \dots, M\}\}$ where $z_n(X) = C(X) + \frac{1}{n}\sum_{j=1}^{n} H(X, \hat{c}^j)$.

With the first stage solutions $X^1, \dots, X^M$ being obtained, the optimal values are denoted $z_m^i = z_m(X^i) = C(X^i) + \frac{1}{m}\sum_{j=1}^{m} H(X^i, c^{ij})$. The average value over all sets of scenarios, $\bar{z}_m = \frac{1}{M}\sum_{i=1}^{M} z_m^i$ is a statistical estimate for a lower bound on the optimal value of the true problem.

For the larger set of n scenarios, which can be regarded as a benchmark scenario set representing the true distribution (see [12]), the cost $z_n(X^i)$ of each solution $X^i, i \in \{1, \dots, M\}$ is computed as well as $X^* = argmin\{z_n(X^i) : i \in \{1, \dots, M\}\}$. The best value, $z_n(X^*)$, is a statistical estimate for an upper bound on the optimal value. The estimated optimality gap (GAP) is given by $GAP = z_n(X^*) - \bar{z}_m$.

When employing a scenario generation method it is desirable that no matter which set of scenarios is used, by solving the two-stage model, one obtains approximately the same value for the optimal solution. This is named as stability requirement conditions in [12]. Here we evaluate stability (following [21]) through the computations of the variances:

$$\hat{\sigma}^2_{z_n(X^*)} = \frac{1}{(n-1)n} \sum_{j=1}^{n} \left(C(X^*) + H(X^*, \hat{c}^j) - z_n(X^*) \right)^2, \tag{35}$$

$$\hat{\sigma}^2_{\bar{z}_m} = \frac{1}{(M-1)M} \sum_{i=1}^{M} (z^i_m - \bar{z}_m)^2, \tag{36}$$

where $\hat{\sigma}^2_{\bar{z}_m}$ is the variance between samples and $\hat{\sigma}^2_{z_n(X^*)}$ is the variance within the larger sample. The estimated variance of the estimated optimality gap is

$$\hat{\sigma}^2_G = \hat{\sigma}^2_{z_n(X^*)} + \hat{\sigma}^2_{\bar{z}_m}.$$

4.2 Optimization process

To solve the model (1) - (34) for a set of scenarios Ω, we first solve to optimality a master problem including only one scenario. Since a feasible solution to the first stage can be completed with a feasible solution to the second stage for each scenario, the resulting values for the first stage decision variables are feasible for the complete problem with all scenarios. However, we need to check whether the solution is optimal for the complete model. To do that we check, for each scenario, whether there is backlogged demand when the deliveries are made as early as possible. If such a scenario with backlogged demand is found, we add to the master problem additional variables and constraints (which are implied by the time constraints and inventory constraints) enforcing the backlogged demand to be counted in the objective function. Then the revised master problem is solved again, and the process is repeated until all the optimality constraints are satisfied. Hence, as in the L-shaped method, the master problem initially disregards the recourse cost, and an improved estimation of the recourse cost is gradually added to the master problem by solving optimality subproblems and adding corresponding cuts. The algorithm may also be terminated if the additional recourse cost added in an iteration is less than a given small amount ϵ. A formal description of this process is given below.

Algorithm 1 Optimization procedure for an input set of scenarios Ω.

```
Choose a scenario c ∈ Ω
Solve the master problem with one scenario, c
while There are new violated optimality cuts and a change in the objective function greater
than ε do
    Add all the violated optimality cuts
    Solve again the master problem with the new cuts
end while
```

Next we explain how separation of constraints imposing backlog for each scenario (optimality cuts) is done in each iteration.

The backlog variables are bounded as follows:

$$r_{imkc} \geq R_{ik}t_{imc} - S_{ik}^{O} - \sum_{n\leq m-1}\sum_{v\in V} q_{invk}, (i,m) \in S^A, k \in K : J_{ik} = -1, c \in \Omega, \quad (37)$$

$$r_{imkc}^{E} \geq R_{ik}t_{imc}^{E} - S_{ik}^{O} - \sum_{n\leq m}\sum_{v\in V} q_{invk}, (i,m) \in S^A, m < \mu_i, k \in K : J_{ik} = -1, c \in \Omega, \quad (38)$$

$$r_{i\mu_i kc}^{E} \geq R_{ik}t_{imc}^{E} + R_{ik}t_{ic}^{+} + \underline{S}_{ik} - S_{ik}^{O} - \sum_{n\leq m}\sum_{v\in V} q_{invk}, i \in N, k \in K : J_{ik} = -1, c \in \Omega. \quad (39)$$

Constraints (37) - (38) are implied by (28) - (30) (adding alternately (30) and (29) from (i,m) to $(i,1)$ and then (28)), and from the non-negativity requirements on the inventory variables (33). Constraints (39) are implied by (28) - (30) and by (32).

The minimum backlog occurs when the time variables t_{imc} and t_{imc}^{E} are set to the earliest feasible times. Once these variables are defined, separation over (37) - (38) is trivial since the right hand side is fixed. So we focus now on finding tight bounds for the time variables. First observe that the starting and ending times of each operation are established either from the (maximum) inventory levels (inventory constraints) or from the duration of the several operations the ships perform (time constraints). In the first case we need to ensure that the inventory capacity is not exceeded. Hence we have:

$$t_{imc} \geq \frac{S_{ik}^{O} + \sum_{n\leq m-1}\sum_{v\in V} q_{invk} - \overline{S}_{ik}}{R_{ik}}, \qquad (i,m) \in S^A, k \in K : J_{ik} = -1, c \in \Omega, \quad (40)$$

$$t_{imc}^{E} \geq \frac{S_{ik}^{O} + \sum_{n\leq m}\sum_{v\in V} q_{invk} - \overline{S}_{ik}}{R_{ik}}, \qquad (i,m) \in S^A, k \in K : J_{ik} = -1, c \in \Omega. \quad (41)$$

Constraints (40) and (41) follow from (28) - (32). For a given feasible solution for the first stage, the right hand sides of (40) and (41) are constant.

For the second case, the time variables are determined from the time constraints (21) - (27). For a feasible solution of the first stage, and for each scenario, most of the constraints (21) - (27) are not tight and many variables do not need to be considered. We can see that the t_{imc}^{E}-variables are bounded by (21) while the t_{imc}-variables are bounded by (22) (from the end time of the visit to the same port) and by (23) (from the last ship operation). These cases can be represented in a network $\mathcal{N} = (P, A, W)$, where P is the set of nodes, A is the set of arcs and W is the set of weights. The set of nodes P is given by the origin of each ship, represented by O_v, a node (i,m) representing the starting time of each port visit and a node $\overline{(i,m)}$ representing the end time of the visit. Each arc in A corresponds to a routing variable set to one. That is, there is an arc from node O_v to node (i,m) if $x_{imv}^{O} = 1$, and there is an arc from node $\overline{(i,m)}$ to node (j,n) if $x_{imjnv} = 1$ for some v. The arcs have weights $T_{ivc}^{O} + T_{imc}^{W}$ and $T_{ijvc} + T_{jnc}^{W}$, respectively. There is an arc from node (i,m) to node $\overline{(i,m)}$ with weight $\sum_{v\in V}\sum_{k\in K_v} T_{ik}^{S} o_{imvk} + \sum_{v\in V}\sum_{k\in K_v} T_{ik}^{Q} q_{imvk}$, and there is an arc from node $\overline{(i,m)}$ to node $(i,m+1)$ with weight T_i^B. Finally, we consider an arc from O_v to each node visited by ship v. The weight from O_v to (i,m) is given by the right hand side of (40) and the weight from O_v to $\overline{(i,m)}$ is given by the right hand side of (41).

The weight of each path from one origin to a node gives a lower bound for the time variable corresponding to that node. Hence the earliest time associated to a node corresponds to the weight of the longest path from one origin to that node (one can always establish an artificial origin which is linked to all ship origins O_v and with null weight). Since the graph is acyclic, finding the longest path to each node can be done in polynomial time. However, for this particular graph, it is easy to derive a linear labeling correcting algorithm.

The time variables can then be restricted using these paths or sub-paths. For each (sub)path $\Pi_{(j,n)}^{(i,m)}$, from visit (j,n) to visit (i,m) of a ship v, we define the set of nodes (port visits) as $\mathcal{N}(\Pi_{(j,n)}^{(i,m)})$ and the set of arcs as $\mathcal{A}\left(\Pi_{(j,n)}^{(i,m)}\right)$. Let (i_v, m_v) denote the first visit of ship v after leaving the origin.

If the earliest time for a visit $(i,m) \in S^A$ is determined only by the schedule of operations for a given ship $v \in V$, then t_{imc} and t_{imc}^E are restricted as follows:

$$\begin{aligned}
t_{imc} \geq & \sum_{(\ell,u)\in\mathcal{N}(\Pi_{(i_v,m_v)}^{(i,m)})} T_{\ell uc}^W + \sum_{(\ell,u)\in\mathcal{N}(\Pi_{(i_v,m_v)}^{(i,m)})\setminus\{(i,m)\}} \sum_{k\in K} \left(T_{\ell k}^S o_{\ell uvk} + T_{\ell k}^Q q_{\ell uvk}\right) \\
& + T_{i_v vc}^O + \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi_{(i_v,m_v)}^{(i,m)})} T_{\ell tvc} \\
& - T\left(1 + \mid \mathcal{A}(\Pi_{(i_v,m_v)}^{(i,m)}) \mid - x_{i_v m_v v}^O - \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi_{(i_v,m_v)}^{(i,m)})} x_{\ell utwv}\right), \qquad (42)
\end{aligned}$$

$$\begin{aligned}
t_{imc}^E \geq & \sum_{(\ell,u)\in\mathcal{N}(\Pi_{(i_v,m_v)}^{(i,m)})} T_{\ell uc}^W + \sum_{(\ell,u)\in\mathcal{N}(\Pi_{(i_v,m_v)}^{(i,m)})} \sum_{k\in K} \left(T_{\ell k}^S o_{\ell uvk} + T_{\ell k}^Q q_{\ell uvk}\right) \\
& + T_{i_v vc}^O + \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi_{(i_v,m_v)}^{(i,m)})} T_{\ell tvc} \\
& - T\left(1 + \mid \mathcal{A}(\Pi_{(i_v,m_v)}^{(i,m)}) \mid - x_{i_v m_v v}^O - \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi_{(i_v,m_v)}^{(i,m)})} x_{\ell utwv}\right). \qquad (43)
\end{aligned}$$

Validity of (42) and (43) is implied by (21) - (23). In Appendix we provide a list of the remaining inequalities defined for each possible subpath.

The overall separation procedure for each iteration works as follow:

Algorithm 2 Separation procedure

Construct the network $\mathcal{N} = (P, A, W)$
Determine the longest path from the origin to each node
Associate the corresponding time variables to each node
Set the backlog variables to the minimum value using (37)-(39)
for each node **do**
  if the corresponding backlog variable has value strictly greater than its value in the current solution **then**
    add the inequality (37)-(39) determining its value
    add the time constraints (40)-(43) corresponding to the weight of the longest path
    (Use subpaths of each ship that are not contained in other subpaths of the same ship in the critical path)
  end if
end for

Example 4.1. *Consider an instance with 2 ships, v_1 and v_2, and 3 ports and assume that there is only one scenario. Hence we omit the corresponding scenario index from all variables and parameters. Let the paths resulting from the first stage solution be $x^O_{11v_1} = x_{1132v_1} = 1$ and $x^O_{21v_2} = x_{2131v_2} = x_{3112v_2} = 1$. Assume the weights of the arcs are those given in Figure 2. For instance, $T^B_1 = T^B_3 = 0.5$, $\sum_{k\in K}\left(T^S_{1k} o_{11v_1k} + T^Q_{1k} q_{11v_1k}\right) = 1$ and $T_{13v_1} = 6$.*

For simplicity we omit arcs with weights resulting from (40) *and* (41)*.*

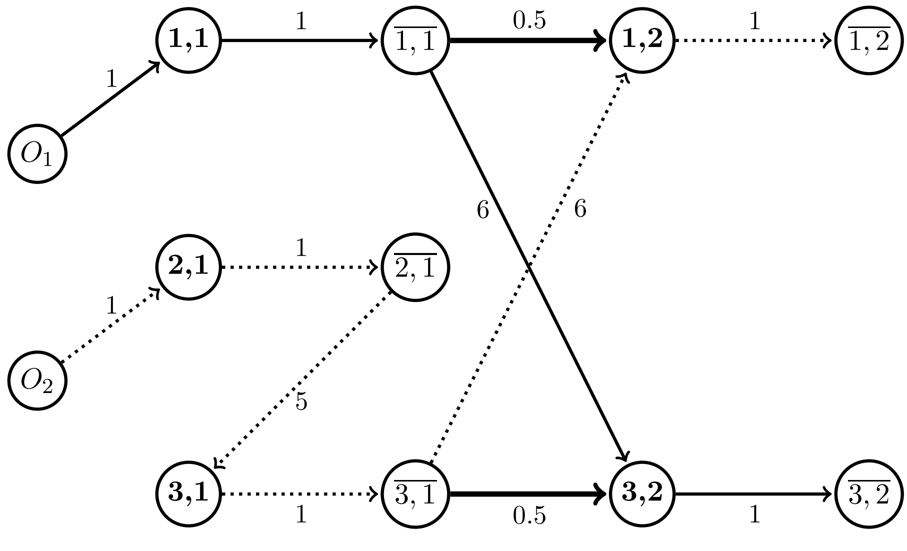

Figure 2: Example of a graph G for a set of three ports and two ships.

We can see that $t_{11} = 1, t^E_{11} = 2, t_{21} = 1, t^E_{21} = 2, t_{31} = 7, t^E_{31} = 8, t_{12} = 14, t^E_{12} = 15, t_{32} = 8.5, t^E_{32} = 9.5$.

Suppose there is backlog at nodes $(1, 2)$ *and* $(3, 2)$. *In addition to the inequalities* (37) *for* $(1, 2)$ *and* $(3, 2)$ *defining the lower bound on the backlog, the following inequalities, corresponding to the* *critical paths to nodes* $(1,2)$ *and* $(3,2)$ *are added to limit the time variables:*

$$\begin{aligned}
t^E_{31} &\geq \sum_{k\in K}\left(T^S_{2k}o_{21v_2k}+T^Q_{2k}q_{21v_2k}\right)+\sum_{k\in K}\left(T^S_{3k}o_{31v_2k}+T^Q_{3k}q_{31v_2k}\right)\\
&\quad +T^O_{2v_2}+T_{23v_2}-T(2-x_{021v_2}-x_{2131v_2}),\\
t_{12} &\geq \sum_{k\in K}\left(T^S_{2k}o_{21v_2k}+T^Q_{2k}q_{21v_2k}\right)+\sum_{k\in K}\left(T^S_{3k}o_{31v_2k}+T^Q_{3k}q_{31v_2k}\right)\\
&\quad +T^O_{2v_2}+T_{23v_2}+T_{31v_2}-T(3-x_{021v_2}-x_{2131v_2}-x_{3112v_2}),\\
t_{32} &\geq t^E_{31}+T^B_3.
\end{aligned}$$

5 Stochastic times and sample scenarios generation

In the SMIRP problem, the sailing and waiting times at ports are assumed to be random, following known probability distributions. We now describe the distributions used and how scenarios are generated for the stochastic programming model.

For the sailing times we assume that there are three potential events that affect all the sailing times simultaneously. These correspond to "good weather", "moderate weather" and "bad weather". For good weather, the sailing times are obtained directly from the sailing distance and the ship speed. For moderate weather, the sailing times are 1.5 times the corresponding sailing times in good weather, and for bad weather the sailing times are 2.0 times those in good weather. From the historical data for the season we are considering, a probability is associated to each event.

Contrary to the sailing times, where the weather usually affects all the islands simultaneously, waiting times due to port occupancy depend only on the port. For each visit to each port, we assume that the random variable indicating whether the port is occupied or not follows a Bernoulli distribution with parameter $p \in [0,1]$ (p is the probability of the port being occupied). If the port is occupied then the random variable W indicating the waiting time is given by a truncated exponential distribution

$$F(w)=\begin{cases}0, & w<0,\\ (1-e^{-\lambda w})/A, & 0\leq w\leq M,\\ 1, & w>M,\end{cases}$$

where $A = 1-e^{-M\lambda}$, and λ is such that the expected value of waiting time is $1/\lambda - \frac{Me^{-M\lambda}}{A}$, and M represents the maximum waiting time. Parameters p, λ, and M are obtained from historical data.

The weather events are trivially generated using the given probabilities. For each visit to each port the waiting times are randomly generated as follows: let $p \in [0,1]$ be the probability of the port being occupied. Generate an uniform random variable $U_1 \in [0,1]$. If $U_1 > p$ we assume that the port is not occupied. Otherwise we randomly generate a waiting time from the truncated exponential distribution using the inverse transformation method. The waiting time is given by

$$W=\begin{cases}\frac{ln(1-AU_2)}{-\lambda}, & \text{if } U_1\leq p;\\ 0, & \text{if } U_1>p.\end{cases}$$

where U_2 is an uniform random variable, $U_2 \in [0,1]$.

To generate the set of scenarios, Ω, we first fix the number of scenarios $n = \mid \Omega \mid$ *a priori*. Then each scenario is generated separately, first by generating the sailing times at random and then by generating a random waiting time for each port visit.

6 Computational results

In this section we report the results from the computational experimentation conducted to test the stochastic model. All computations were performed using the optimization software Xpress Optimizer Version 20.00.05 with Xpress Mosel Version 3.0.0, on a computer with processor Intel Core 2 Duo 2.2GHz and with 4GB of RAM. In Algorithm 1 we use $\epsilon = 0.01$. Ten real-world instances are used in the testing, considering two different ships, seven ports, four products, and a time horizon of eight days. The instances differ on the initial inventory levels.

First we test the effectiveness of the decomposition method, through a comparison by solving the full stochastic programming model directly using commercial software. Then, we test the sample average approximation method using the decomposition method. Finally we compute estimations of the Value of the Stochastic Solution and the Expected Value of Perfect Information.

6.1 Effectiveness of decomposed model

To test the effectiveness of the decomposed model we compared its performance with the use of Xpress Optimizer to directly solve the stochastic programming model with 10 scenarios. The results are reported in Table 1. The column "Opt" gives the optimal values, the columns "Nodes" indicate the number of branch and bound nodes, the columns "Seconds" report the running time in seconds to solve the instance. For the decomposed model we report additionally the number of cuts added in the column "Ncuts" and the number of iterations in the column "Iterations", that is, the number of times we solve the separation problem to add backlog and time constraints.

Table 1: Effectiveness of the L-shaped method

		full model		**decomposed model**			
Instance	Opt	Nodes	Seconds	Nodes	Seconds	Ncuts	Iterations
1	16210	5888	1498	3503	390	20	3
2	17610	20292	5397	16436	1061	24	4
3	18500	8495	2111	3863	434	65	3
4	17248.6	9253	1644	5377	526	78	4
5	15410	8177	2284	5356	384	18	3
6	18576.8	42774	7799	7247	854	32	4
7	15362.3	20720	4546	6658	603	45	4
8	17008	27740	5564	7558	740	28	4
9	13330	1911	362	1462	146	16	3
10	14550	46407	9200	3821	351	25	4
Average	16380.57	19165.7	4040.5	6128.1	548.9	35.1	3.6

As expected, the running times of the decomposition method are much lower than the running times obtained by solving the complete model. Additionally, we can see that the number of times the separation problem is called is at most 4 and few cuts are added.

6.2 Testing different sizes of sets of scenarios

Next we follow the solution approach described in Section 4, see [21]. Each instance is solved for M independent sets of scenarios, each set i containing m scenarios.

We conducted tests for $m = 10$ and $m = 50$. In all cases we consider $M = 10$ and the solutions are evaluated using a bigger set of $n = 1000$ scenarios. For each value of m we give two tables (Tables 2 and 3 for $m = 10$ and Tables 4 and 5 for $m = 50$). In the first table we present, for each instance, $z_n(X^*)$, $\bar{z}_m$, GAP, $\hat{\sigma}_{z_n(X^*)}$, $\hat{\sigma}_{\overline{z}_m}$, $\hat{\sigma}_G$. In the second table we give, for each instance, the running time to solve the M problems (one problem for each set of scenarios of size m) using the decomposition method, "Seconds M", and the time to compute $z_n(X^k), k \in \{1, \ldots, M\}$ "Seconds n", the average number of iterations, "Iterations", to solve the M problems, that is, the average number of times we solve the separation problem, and the average number of cuts (37)-(39) added, "Cuts".

Table 2: Bounds and variances for $m = 10$.

Instance	$z_n(X^*)$	$\bar{z}_m$	**GAP**	$\hat{\sigma}_{z_n(X^*)}$	$\hat{\sigma}_{\overline{z}_m}$	$\hat{\sigma}_G$
1	16956.8	16358	598.8	24.1	76.7	80.4
2	19080.3	18516.9	563.4	47.2	173.8	180.1
3	21150.9	19660.2	1490.7	95.6	265.1	281.8
4	19613.9	18750.8	863.1	198.4	293.1	353.9
5	18813.2	16658.5	2154.7	72.3	194.2	207.3
6	21182.1	19743.3	1438.8	104.7	210.7	235.3
7	16694.8	16509.5	185.3	76.6	196.8	211.1
8	19325.2	18664.2	661	71.0	227.8	238.6
9	14335.6	14139	196.6	115.2	238.1	264.5
10	17636.8	16721.6	915.2	127.0	324.5	348.5
Average	18479.0	17572.2	906.8	93.2	220.1	240.1

Table 3: Times, average number of iterations and average number of cuts for $m = 10$.

Instance	**Seconds M**	**Seconds n**	**Iterations**	**Cuts**
1	77	35	3	16.4
2	180.5	38.5	4	16.4
3	95.57	45.7	3	16.4
4	104.9	42.5	3.4	18.4
5	118.3	41.2	3	15.2
6	181.7	50.9	3	14.4
7	133.5	29.3	3	13.9
8	117.7	54.3	3.2	19.6
9	37.1	42.9	3	20.2
10	123.1	40.7	3	16.3
Average	116.9	42.1	3.2	16.7

We can see that increasing m, the cost of the selected solution decreases in average by 2.8%. Also, the standard deviation $\hat{\sigma}_{\overline{z}_m}$ and the GAP have a little reduction. The price to pay for the improvement of the solution and reduction of variability is an increase in the average running times. The running time is, on average, approximately 2 minutes for $m = 10$, and increases to 10 minutes for $m = 50$.

Table 4: Bounds and variances for $m = 50$.

Instance	$z_n(X^*)$	$\bar{z}_m$	**GAP**	$\hat{\sigma}_{z_n(X^*)}$	$\hat{\sigma}_{\bar{z}_m}$	$\hat{\sigma}_G$
1	16498.1	16352.4	145.7	11.5	44.2	45.7
2	19080.3	18375.8	704.5	47.2	219.3	224.3
3	20839.2	19542.5	1296.7	97.6	142.0	172.3
4	19401.7	18027.2	1374.5	198.4	241.2	312.3
5	18490.2	16160	2330.2	66.2	119.3	136.4
6	20177.1	19651.1	526	104.5	96.4	142.2
7	16358.8	16204.4	154.4	76.6	98.5	124.8
8	18148.3	18015.3	133	71.0	198.2	210.6
9	13877.7	13876.4	1.3	115.2	116.5	163.8
10	16875	16336.1	538.9	127.0	286.4	313.3
Average	17974.6	17254.1	720.5	91.5	156.2	184.5

Table 5: Times, average number of iterations and average number of cuts for $m = 50$.

Instance	**Seconds M**	**Seconds n**	**Iterations**	**Cuts**
1	426.5	51	3.3	18.8
2	838.2	64.8	3	16.5
3	658.3	59.4	3.1	18.9
4	682.8	57.8	3	20.1
5	627.2	63.9	3	16.8
6	751.9	50.2	4.2	16.8
7	573.3	76.1	3	16.3
8	791	49.5	3	19.8
9	174.5	51.5	4	18.8
10	524.3	56	3	17.8
Average	604.8	58.0	3.3	18.1

6.3 Importance of a stochastic approach

To evaluate the importance of the stochastic approach we compute estimations of the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI). The results are given in Table 6. To compute the VSS we solve the model with one scenario, where the stochastic parameters are set to their expected values. We used the sample average values (considering the larger sample), which are very similar to the theoretical expected values. Solving this deterministic model we obtain the well known expected value solution. The cost of this solution is given in column "EVS". In column $z_n(X^*)$ we give the corresponding value for $m = 50$, and in column "VSS" we give an estimation of the Value of the Stochastic Solution which is the difference between EVS and $z_n(X^*)$. In column "PI" we give the average value of the $n = 1000$ deterministic models, one for each scenario, and in column "EVPI" we give an estimation of the Expected Value of Perfect Information which is the difference $z_n(X^*) - PI$.

We can see, from Table 6, the gains for using stochastic programming instead of the deterministic model based on expected values are in general very high. In average, the expected value of the best solution is only 9% above the Expected Value of Perfect Information.

Table 6: Estimating the VSS and EVPI

Instance	EVS	$z_n(X^*)$	VSS	PI	EVPI
1	42049.2	16498.1	25551.1	16210.1	288.0
2	33020.1	19080.3	13939.8	17620.2	1460.1
3	39871.2	20839.2	19032	18572.6	2266.6
4	49582.5	19401.7	30180.8	17285.9	2115.8
5	58053.6	18490.2	39563.4	15461.7	3028.5
6	41503.3	20177.1	21326.2	18942.0	1235.1
7	32256.6	16204.4	16052.2	15497.8	706.6
8	64144	18148.3	45995.7	17023.8	1124.5
9	41125.7	13877.7	27248	13354.0	523.7
10	25623.9	16875	8748.9	15066.9	1808.1
Average	42723.01	17959.2	24763.8	16503.5	1455.7

7 Conclusions

We presented a two-stage stochastic programming model with recourse for a maritime inventory routing problem where sailing times and port times are random. The model has the property that, for each scenario, a feasible solution to the first stage can always be completed with a feasible solution to the second stage. We proposed a decomposition method where, for a given first stage solution, optimality is checked for the complete model through an efficient separation method.

Ten instances based on real data are solved using the sample approximation method. Computational tests have shown the effectiveness of the decomposition method, and the importance in the use of stochastic programming instead of a deterministic approach.

Acnowledgements

The work of the first and third authors was supported in part by *FEDER* funds through *COMPETE*–Operational Programme Factors of Competitiveness ("Programa Operacional Factores de Competitividade") and by Portuguese funds through the *Center for Research and Development in Mathematics and Applications (CIDMA)* and the Portuguese Foundation for Science and Technology ("FCT–Fundação para a Ciência e a Tecnologia"), within project PEst-C/MAT/UI4106/2011 with COMPETE number FCOMP-01-0124-FEDER- 022690. The second and forth authors were supported financially from the Research Council of Norway through the DOMinant II-project.

References

[1] Agra A., Christiansen M., Delgado A., Mixed integer formulations for a short sea fuel oil distribution problem. Transportation Science, 47:108-124, 2013.

[2] Agra A., Christiansen M., Delgado A., Simonetti L., Hybrid heuristics for a maritime short sea inventory routing problem (submitted to journal), 2013.

[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] Azaron A., Kianfar F., Dynamic shortest path in stochastic dynamic networks: Ship routing problem. European Journal of Operational Research, 144:138–156, 2003.

[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., Nygreen B., Ronen D., Ship routing and scheduling in the new millennium. European Journal of Operational Research, 228:467–483, 2013.

[8] Christiansen M., Nygreen B., Robust inventory ship routing by column generation. In: Desrosiers G. and Solomon M.M. (Eds.), *Column Generation. Springer-Verlag*, New York, 197–224, 2005.

[9] Halvorsen-Weare E.E., Fagerholt K., Robust supply vessel planning. In J. Pahl, T. Reiners, and S. Voß, editors, *Network Optimization*, volume 6701 of *Lecture Notes in Computer Science*, 559–573. Springer, Berlin, 2011.

[10] 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.

[11] Higle J., Stochastic Programming: Optimization when uncertainty matters. In: Greenberg H.J.and Cole Smith J. (Eds.), *Tutorials in operations research. INFORMS*, New Orleans, 30–53, 2005.

[12] Kaut M., Wallace S., Evaluation of scenario-generation methods for stochastic programming, Pacific Journal of Optimization, 3:257–271, 2007.

[13] Kenyon A.S., Morton D.P., Stochastic vehicle routing with random travel times. Transportation Science, 37:69–82, 2003.

[14] Lambert V., Laporte G., Louveaux F., Designing collection routes through bank branches. Computers and Operations Research, 20:783–791, 1993.

[15] Laporte, G., Louveaux F., Mercure F., The vehicle routing problem with stochastic travel times. Transportation Science, 26:161–170, 1992.

[16] Lo H.K., McCord M.R., Adaptive ship routing through stochastic ocean currents: General formulations and empirical results. Transportation Research, Part A, 32:547–561, 1998.

[17] 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.

[18] 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. CRC Press*, Boca-Raton, 137–166, 2006.

[19] 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. CRC Press*, Boca-Raton, 167–194, 2006.

[20] Teng S.Y., Ong H.L., Huang H.C., An integer L-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times. Asia-Pacific Journal of Operational Research, 21:241–257, 2004.

[21] Verweij B., Ahmed B., Kleywegt A.J, Nemhauser G., Shapiro A., The sample average approximation method applied to stochastic routing problems: a computational study, Computational Optimization and Applications, 289–333, 2003.

Appendix

The following inequalities, for each $(i,m) \in S^A$ and $v \in V$, are implied by (21) - (23):

$$\begin{aligned} t_{imc} \;\geq\; & t_{jnc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})\setminus\{(j,n)\}} T^W_{\ell uc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})\setminus\{(i,m)\}} \sum_{k\in K} \left(T^S_{\ell k} o_{\ell uvk} + T^Q_{\ell k} q_{\ell uvk}\right) \\ & + \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} T_{\ell tvc} - T\left(\mid \mathcal{A}(\Pi^{(i,m)}_{(j,n)}) \mid - \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} x_{\ell utwv}\right), \end{aligned} \tag{44}$$

$$\begin{aligned} t_{imc} \;\geq\; & t^E_{jnc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})\setminus\{(j,n)\}} T^W_{\ell uc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})\setminus\{(j,n),(i,m)\}} \sum_{k\in K} \left(T^S_{\ell k} o_{\ell uvk} + T^Q_{\ell k} q_{\ell uvk}\right) \\ & + \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} T_{\ell tvc} - T\left(\mid \mathcal{A}(\Pi^{(i,m)}_{(j,n)}) \mid - \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} x_{\ell utwv}\right), \end{aligned} \tag{45}$$

$$\begin{aligned} t^E_{imc} \;\geq\; & t_{jnc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})\setminus\{(j,n)\}} T^W_{\ell uc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})} \sum_{k\in K} \left(T^S_{\ell k} o_{\ell uvk} + T^Q_{\ell k} q_{\ell uvk}\right) \\ & + \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} T_{\ell tvc} - T\left(\mid \mathcal{A}(\Pi^{(i,m)}_{(j,n)}) \mid - \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} x_{\ell utwv}\right), \end{aligned} \tag{46}$$

$$\begin{aligned} t^E_{imc} \;\geq\; & t^E_{jnc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})\setminus\{(j,n)\}} T^W_{\ell uc} + \sum_{(\ell,u)\in\mathcal{N}(\Pi^{(i,m)}_{(j,n)})\setminus\{(j,n)\}} \sum_{k\in K} \left(T^S_{\ell k} o_{\ell uvk} + T^Q_{\ell k} q_{\ell uvk}\right) \\ & + \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} T_{\ell tvc} - T\left(\mid \mathcal{A}(\Pi^{(i,m)}_{(j,n)}) \mid - \sum_{(\ell,u,t,w)\in\mathcal{A}(\Pi^{(i,m)}_{(j,n)})} x_{\ell utw}\right). \end{aligned} \tag{47}$$