Flexible Solutions to Maritime Inventory Routing Problems with Delivery Time Windows

Chengliang Zhang   George Nemhauser   Joel Sokol

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology, Atlanta, Georgia 30332

czhang85@gatech.edu   george.nemhauser@isye.gatech.edu   jsokol@isye.gatech.edu

Myun-Seok Cheon   Ahmet Keha

ExxonMobil Research and Engineering Company, Annandale, New Jersey 08801

myun-seok.cheon@exxonmobil.com   ahmet.b.keha@exxonmobil.com

This paper studies a Maritime Inventory Routing Problem with Time Windows (MIRPTW) for deliveries with uncertain disruptions. We consider disruptions that increase travel times between ports and ultimately affect the deliveries in one or more time windows. The objective is to find flexible solutions that can withstand unplanned disruptions. We propose a Lagrangian heuristic algorithm of obtaining flexible solutions by introducing auxiliary soft constraints that are incorporated in the objective function with Lagrange multipliers. To evaluate the flexibility of solutions, we build a simulator that generates disruptions and recovery solutions. Computational results show that by incurring a small increase in initial cost (sometimes zero), our robust planning strategies generate solutions that are often significantly less vulnerable to potential disruptions. We also consider the effect of lead time in being able to respond to the disruptions.

Key words: Inventory routing, Uncertainty, Simulation, Lagrangian heuristic.

1 Introduction

The classical Maritime Inventory Routing Problem with Time Windows (MIRPTW) is to find an optimal routing plan that minimizes the total cost of transportation, while satisfying inventory constraints and contractual delivery constraints. However, in practice, unpredictable disruptions may affect the execution of an optimal deterministic plan. Among all the uncertain factors in maritime transportation, one of the most common ones is that travel times are affected by weather conditions. Focusing on this type of uncertainty, we consider MIRPTW with unpredictable disruptions.

Various definitions and approaches for schedule robustness have appeared in the literature. Robust optimization (Ben-Tal et al. (2009)) is one modeling framework for dealing with uncertain data in optimization. However, the worst-case assumption of robust optimization can lead to solutions that are too conservative. On the other hand, because of the large number of uncertain scenarios that need to be considered, stochastic programming (Shapiro et al. (2009)) has computational limitations for this class of problems. Fischetti and Monaci (2009) propose a general heuristic scheme for robustness called Light Robustness where a set of slack variables is used to measure an estimate of the solution robustness and their sum is minimized in the objective function. In this study, we focus on generating flexible solutions with limited vulnerability to unpredictable disruptions, and use a different approach for dealing with the uncertainty. After analyzing problem characteristics that may provide robustness, we quantify them as soft constraints that are incorporated in the objective function with Lagrange multipliers. We use a subgradient algorithm to find candidate solutions to evaluate. Furthermore, to evaluate the flexibility of schedules, we build a simulator that generates disruptions and recovery solutions. By simulating various disruption events, we show that the actual operational costs in case of disruptions can be significantly reduced when flexible plans are implemented. To the best of our knowledge, only Cacchiani et al. (2012) discusses this kind of approach for dealing with robustness in the literature. They propose a Lagrangian heuristic for solving a robust train timetabling problem. The process collects a set of “Pareto optimal” heuristic solutions, and the robustness of a solution is evaluated by calculating a predefined measure.

Christiansen et al. (2004), Christiansen et al. (2007) and Papageorgiou et al. (2012) give comprehensive reviews of maritime inventory routing problems. However, there are only a few studies that deal with robust planning in the shipping industry. Christiansen and Fagerholt (2002) study a multi-ship pickup and delivery problem with soft time windows. They design robust schedules that are less likely to result in ships staying idle at ports during weekends by imposing penalty costs for arrivals at risky times. Also motivated by uncertainties in maritime transportation, Agra et al. (2012) and Agra et al. (2013) investigate a vehicle routing problem with time windows where travel times are uncertain and belong to a predetermined polytope. A robust optimization framework is used to find routes that are feasible for all values of the travel times in the uncertainty polytope. Similarly, the robust optimization framework is applied in Alvarez et al. (2011) to solve a multi-period fleet sizing and deployment problem with uncertainty in price and demand. A simulation study for a liquefied natural gas (LNG) ship routing problem with uncertainty in sailing time and production rate is presented in Halvorsen-Weare et al. (2013), and several robustness strategies are discussed in the paper.

More work has been done on stochastic airline scheduling problems. Various studies of robust scheduling in the airline industry can be found in Ageeva (2000), Rosenberger et al. (2003), Rosenberger et al. (2004), Schaefer et al. (2005), Lan et al. (2006), Shebalov and

Klabjan (2006), Smith and Johnson (2006), Yen and Birge (2006) and Chiraphadhanakul (2010).

The main contributions of this paper are: (i) a general Lagrangian heuristic scheme to deal with robustness where soft constraints are used to promote solution characteristics that lead to robustness, and (ii) a simulator that evaluates the quality of the solutions found by the Lagrangian heuristic algorithm and determines the cost of achieving the robustness.

The remainder part of the paper is organized as follows. Section 2 provides a description of the problem and the mathematical model formulation. Section 3 presents soft constraints that are used to enhance robustness and proposes the Lagrangian heuristic scheme to generate flexible solutions. Section 4 discusses the simulator, random disruptions and the recovery model. We report computational results in Section 5.

2 Problem description

The MIRPTW studied in this paper is motivated by a long-term planning problem in maritime transportation. The goal is to find an optimal schedule for a heterogeneous pool of ships that deliver a product from a set of loading ports to a set of discharging ports such that constraints related to inventory storage at loading ports and contractual obligations at discharging ports are satisfied.

We assume there is a set of loading ports denoted by $\mathcal{J}^{\mathcal{L}}$. The i-th port has a constant production rate p^i of a single product, an initial inventory level I_0^i and a specified storage capacity F^i. On the discharging side, we consider a set of discharging ports denoted by $\mathcal{J}^{\mathcal{D}}$ where the j-th port has a set $\mathcal{K}_j$ of time windows. For the k-th time window at discharging port j, it is assumed that u_{jk} and v_{jk} are the first and the last day in the time window respectively, and q_{jk} is the committed delivery quantity. The contractual agreements are to deliver quantity q_{jk} of product in each specified time window $[u_{jk}, v_{jk}]$ at each discharging port by using a set of heterogeneous ships denoted by $\mathcal{V}$, each with some load capacity W_v. Let $\mathcal{T}$ be the set of periods in the planning horizon and $\mathcal{J} = \mathcal{J}^{\mathcal{L}} \cup \mathcal{J}^{\mathcal{D}}$ be the set of all ports. We restrict our attention to problems where ships discharge their entire load at one port. However, we do not assume that ships necessarily carry full loads. Therefore, we track the quantities on ships for all periods, and use O_t^v to denote the quantity on ship $v \in \mathcal{V}$ after period $t \in \mathcal{T}$. Also, we assume that multiple ships may combine to serve the same time window. From a planning perspective, we believe this is a rather generic problem that is adaptable to several realistic applications.

One notable difference from most inventory routing and vehicle routing problems considered in the literature is that instead of just providing the distance between each pair of ports, we represent each port by a two-dimensional vector in a coordinate plane. By assuming that ships travel in a straight line between two ports at a constant speed, we can track the geographic locations of ships for all periods. The purpose of doing this is that we are able to adjust the original plan by rerouting some ships en route in case of disruptions. This will be discussed further in Section 4.

Song and Furman (2013) introduce a practical modeling framework for a class of Maritime Inventory Routing Problems (MIRPs), and the model we use in this study shares many features with this proposed framework. The model is constructed on a time-space network. The network has a source node n_0, a sink node n_T and a set $\mathcal{N}$ of regular nodes where each regular node n is a port-time pair (j, t), $j \in \mathcal{J}, t \in \mathcal{T}$. The nodes are shared by all the ships, while each ship has its own travel and demurrage arcs in the network. The travel arcs from node (j_1, t_1) to node (j_2, t_2) represent travel between ports j_1 and j_2, and the demurrage arcs from node (j,t) to node $(j,t+1)$ represent waiting at port j. We use $\mathcal{A}$ to denote the set of all arcs, and $\mathcal{A}^+$ to denote the set of all travel arcs. In addition, the sets of incoming and outgoing travel arcs associated with ship v and node $n=(j,t)$ are denoted by $\mathcal{RS}(j,t,v)^+$ and $\mathcal{FS}(j,t,v)^+$ respectively, while the sets of incoming and outgoing arcs associated with ship v and node $n=(j,t)$ are denoted by $\mathcal{RS}(j,t,v)$ and $\mathcal{FS}(j,t,v)$ respectively.

Let $x_a = 1$ if arc $a \in \mathcal{A}$ is used and $x_a = 0$ otherwise, and f_{nv} be the loading/discharging quantity at node $n \in \mathcal{N}$ by ship $v \in \mathcal{V}$. An arc-flow mixed integer programming model is given by (P):

$$\min \quad \sum_{a \in \mathcal{A}^+} c_a x_a \tag{1}$$

$$s.t. \quad \sum_{a \in \mathcal{FS}(n,v)} x_a - \sum_{a \in \mathcal{RS}(n,v)} x_a = 0, \qquad \forall v \in \mathcal{V},\ \forall n \in \mathcal{N} \tag{2}$$

$$\sum_{a \in \mathcal{FS}(n_0,v)} x_a = 1, \qquad \forall v \in \mathcal{V} \tag{3}$$

$$\sum_{a \in \mathcal{RS}(n_T,v)} x_a = 1, \qquad \forall v \in \mathcal{V} \tag{4}$$

$$I_t^i = I_{t-1}^i + p^i - \sum_{v \in \mathcal{V}} f_{nv}, \qquad 0 \leq I_t^i \leq F^i, \qquad \forall i \in \mathcal{J}^{\mathcal{L}},\ \forall t \in \mathcal{T} \tag{5}$$

$$O_t^v = O_{t-1}^v + \sum_{j \in \mathcal{J}^{\mathcal{L}}} f_{nv} - \sum_{j \in \mathcal{J}^{\mathcal{D}}} f_{nv}, \qquad 0 \leq VI_t^v \leq W_v, \qquad \forall t \in \mathcal{T},\ \forall v \in \mathcal{V} \tag{6}$$

$$\sum_{v \in \mathcal{V}} \sum_{u_{jk} \leq t \leq v_{jk}} f_{nv} = q_{jk}, \qquad \forall j \in \mathcal{J}^{\mathcal{D}},\ \forall k \in \mathcal{K}_j \tag{7}$$

$$f_{nv} \leq W_v \sum_{a \in \mathcal{FS}(j,t,v)} x_a, \qquad \forall j \in \mathcal{J},\ \forall t \in \mathcal{T},\ \forall v \in \mathcal{V}. \tag{8}$$

$$O_t^v \leq W_v(1 - \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a), \qquad \forall v \in \mathcal{V},\ \forall t \in \mathcal{T},\ \forall j \in \mathcal{J}^{\mathcal{D}} \tag{9}$$

$$x_a \in \{0,1\},\ f_{nv} \geq 0,\ \forall a \in \mathcal{A},\ \forall v \in \mathcal{V},\ \forall n \in \mathcal{N}. \tag{10}$$

The objective is to minimize total transportation costs, where c_a is the travel cost associated with arc $a \in \mathcal{A}^+$. (2)-(4) are network flow conservation constraints. (5) and (6) are balance constraints of the product at loading ports and ships respectively. (7) ensures that deliveries are completed within the time windows. (8) states that loading or discharging can occur only when the ship is at port. (9) is the full discharge constraint.

Since a ship might have more time than needed to get from one port to another, there can be some slack in planning solutions. Therefore, as a post-processing procedure to improve the robustness of solutions by giving ships as much time as possible for their next voyage, we reallocate the slacks to force ships to depart as soon as possible by using the modified planning model (MP):

$$\min \qquad g_1(x) = \sum_{a \in \mathcal{A}^+} c_a x_a + \sum_{n=(j,t):j \in \mathcal{J}^{\mathcal{D}}} \varepsilon_n \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(n,v)^+ \cup \mathcal{RS}(n,v)^+} x_a \tag{11}$$

$$s.t. \qquad f_{nv} \leq W_v \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a, \qquad \forall j \in \mathcal{J}^{\mathcal{D}},\ \forall t \in \mathcal{T},\ \forall v \in \mathcal{V}. \tag{12}$$

$$(2) - (10)$$

where $\varepsilon_{(j,t_1)} < \varepsilon_{(j,t_2)}$ if $t_1 < t_2$ and they both belong to the same extended time window (see Fig. 1).

The second term in (11) together with constraints (12) ensures that ships leave immediately once they collect enough inventory at loading ports and finish discharging at discharging ports. Slack reallocation does not affect the routing decisions since we choose epsilon small enough that it is dominated by the transportation costs in (11). Moreover, because slack reallocation also breaks symmetry, (MP) is much easier to solve than (P).

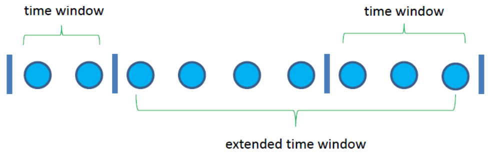

Figure 1: Extended time window

Two techniques are applied to speed up solution times. First, we control the order in which variables are branched by specifying a priority order. Specifically in our model, we create a set of ancillary integer variables defined by

$$s_{vjk} = \sum_{u_{jk} \leq t < v_{jk}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a + \sum_{a \in \mathcal{FS}(j,v_{jk},v)} x_a, \qquad \forall v \in \mathcal{V}, \forall j \in \mathcal{J}^{\mathcal{D}}, \forall k \in \mathcal{K}_j, \tag{13}$$

to represent the number of visits of ship v to time window k at discharging port j, and give them the highest priority.

Secondly, we add a set of enhanced knapsack cover cuts. Since demands are exclusively satisfied from ship deliveries, the inequalities

$$\sum_{v \in \mathcal{V}} W_v s_{vjk} \geq q_{jk}, \qquad \forall j \in \mathcal{J}^{\mathcal{D}},\ \forall k \in \mathcal{K}_j. \tag{14}$$

are valid for the planning model.

3 A Lagrangian heuristic based on relaxing auxiliary soft constraints

In this section, we propose a Lagrangian heuristic scheme to obtain robust solutions to our model. The technique is general and can be applied to any optimization problem concerned with robustness for which *soft constraints* can be identified that are not necessary, but whose satisfaction can aid robustness. Since it may not be possible to satisfy all of the soft constraints while satisfying the hard constraints, we incorporate soft constraints in the objective function with Lagrange multipliers. When we increase the Lagrange multipliers by using a subgradient algorithm, various heuristic solutions of potentially higher planning cost are generated.

For problem (MP), suppose $[u_{jk}, v_{jk}]$ is the k-th time window at discharging port j, where u_{jk} and v_{jk} are the first and the last period within the time window respectively. An m-day soft constraint associated with $[u_{jk}, v_{jk}]$ is defined as

$$\sum_{v \in \mathcal{V}} \sum_{v_{jk}-m+1 \leq t \leq v_{jk}} \sum_{a \in \mathcal{RS}(j,t,v)^+} x_a = 0. \tag{15}$$

It implies that there are no incoming ships in the last m periods before the end of the time window. Combined with the hard constraint (7), (15) makes all ships that are going to serve time window $[u_{jk}, v_{jk}]$ get to the port at least m days in advance. An alternative way of modeling the soft constraints is to replace x_a in (15) with a continuous variable representing the quantity that ship v brings to discharging port j at period t. Since the two approaches give very similar computational results, for the remainder of the paper we use the soft constraints defined by (15).

By associating each soft constraint with a multiplier and including m-day soft constraints ($m = 1, 2, \ldots, r$) for all time windows in the objective function, we have the Lagrangian optimization problem:

$$LR(\theta) \;=\; \min \quad \{g_2(x) = g_1(x) + \sum_{j\in\mathcal{J}^{\mathcal{D}}} \sum_{k\in\mathcal{K}_j} \sum_{m=1}^{r} \theta_{jkm} x_{jkm}\} \tag{16}$$

$$s.t. \qquad (2) - (10),\ (12).$$

where $x_{jkm} = \sum_{v\in\mathcal{V}} \sum_{v_{jk}-m+1\leq t\leq v_{jk}} \sum_{a\in\mathcal{RS}(j,t,v)^+} x_a$ and $\theta_{jkm} \geq 0$ is a Lagrange multiplier. The dual problem $\max_{\theta\geq 0} LR(\theta)$ might be unbounded since if all the soft constraints are included the planning problem is likely to be infeasible. Therefore, we impose an upper bound on θ_{jkm} for all $j \in \mathcal{J}^{\mathcal{D}}, k \in \mathcal{K}_j, m \in \{1, \cdots, r\}$ and have

$$\max_{0\leq\theta\leq M} LR(\theta) \tag{17}$$

where $M = (M^0, \cdots, M^0)$ is a vector and M^0 is a large number.

We use a subgradient algorithm to explore the solution space with the step size at the k-th iteration given by

$$t_k = \frac{\lambda_k(Z^* - LR(\theta^k))}{\|Ax^k - b\|^2} \tag{18}$$

where λ_k is a scalar satisfying $0 \leq \lambda_k \leq 2$, Z^* is an upper bound on problem (17), and $Ax - b$ is the Lagrangian term. Justification of the formula is given in Held et al. (1974). The next proposition gives the optimal value of problem (17), and thus can be used as Z^* in formula (18).

Proposition 1.

$$\max_{0\leq\theta\leq M} LR(\theta) = LR(M). \tag{19}$$

Proof. Proof: Let θ^* be a Lagrange multiplier such that $0 \leq \theta_i^* \leq M^0$ for each i. We assume that x is an optimal solution of $LR(\theta^*)$ and y is an optimal solution of $LR(M)$.

Since $y_{jkm} \geq 0,\ \forall j \in \mathcal{J}^{\mathcal{D}},\ \forall k \in \mathcal{K}_j,\ \forall m = \{1, 2, \ldots, r\}$, we have

$$\begin{aligned} & LR(\theta^*) \\ = \ & g_1(x) + \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^{r} \theta^*_{jkm} x_{jkm} \\ \leq \ & g_1(y) + \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^{r} \theta^*_{jkm} y_{jkm} \\ \leq \ & g_1(y) + M \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} \sum_{m=1}^{r} y_{jkm} \\ = \ & LR(M). \end{aligned}$$

Therefore, $\max_{0 \leq \theta \leq M} LR(\theta) = LR(M)$. □

Algorithm 1 Lagrangian heuristic algorithm for finding robust solutions

```
0. Solve $LR(M)$. Let $U = LR(M)$, and $x_{(M)}$ be an optimal solution.

1. Set $\theta^0 = 0, k = 0, A = \{x_{(M)}\}$.

2. Solve $LR(\theta^k)$. If $LR(\theta^k) \geq (1 - \epsilon)U$ and $x^k \notin A$, $A \leftarrow A \cup \{x^k\}$.

3. If $k = \Omega$ and $A \neq \{x_{(M)}\}$, go to 4.
   Else if $k = \Omega$ and $A = \{x_{(M)}\}$, $A \leftarrow A \cup \{x^\Omega\}$, go to 4.
   Else, update $\theta^{k+1} = \theta^k + t_k(Ax^k - b)$ where $t_k = \frac{2(U - LR(\theta^k))}{\|Ax^k - b\|^2}$, $k \leftarrow k + 1$ and go to 2.

4. Simulate solution $x,\ \forall x \in A$.
```

Now we give the Lagrangian heuristic algorithm for finding robust solutions. In our algorithm, we assume the initial Lagrange multiplier vector $\theta^0 = 0$ and use set A to collect candidate robust solutions. We use U as the upper bound in formula (18), and set $\lambda = 2$ for every iteration. If the gap between the optimal value of $LR(\theta^k)$ and U is no larger than a pre-specified parameter ϵ, solution x^k is labeled as a candidate robust solution. Ω is a pre-specified maximal limit on the number of iterations of the subgradient algorithm, and k denotes the current iteration number. If $x_{(M)}$ is the only candidate robust solution after the last iteration, we include x^{Ω} (the solution in the last iteration) in set A. The final output of the algorithm is a set A that contains all the different candidate robust solutions that will be processed in the simulator presented in the next section to evaluate their actual performance when disrupted.

4 A simulator for evaluating the flexibility of solutions

To evaluate how solutions respond to unexpected disruptions, we built a simulator to study the recovery process from disruptions. The simulator generates random disruptions one by one and reoptimizes the original schedule in each case. To respond to a disruption, we solve a recovery model that incorporates the following three recovery options:

1. *Push-back.*

   If the slacks in the schedule or the time windows are sufficient to absorb the delays, we simply delay the affected routes and do not re-route any ships. Push-back does not increase cost.

2. *Ship re-routing.*

   If necessary, the simulator is able to re-route ships en route to ports different from their original destinations. This recovery option often increases transportation costs.

3. *Spot market.*

   If it is impossible to meet all time window demands by only using the first two options, an expensive spot market acts as an additional supply source in the recovery model.

4.1 Random disruptions

A disruption at a loading port is defined as a four-dimensional vector (t, p, n, l) which means that all the ships that are scheduled to arrive at loading port p at period(s) $t, \cdots, t+n-1$ are delayed to period $t+n$, and the recovery model is able to respond to the disruption at period $t-l$. A disruption at a discharging port is defined as a triple (w, n, l) where w is the affected time window, n is the number of extra extended travel days to time window w on ships that are scheduled to serve w, and l is the lead time of being able to respond to the disruption before it occurs.

By the definition of a disruption, n represents the extent of the disruption and l controls the lead time of when it is possible to respond to the disruption. We now show the obvious result that for a single disruption, the total actual cost is a nonincreasing function of lead time.

Proposition 2. *Suppose $c(p, d)$ is the total actual cost over the entire horizon if p is the planning solution and disruption d occurs during execution. Assume $d_1 = (w, n, l_1)$ (or $d_1 = (t, p, n, l_1)$), $d_2 = (w, n, l_2)$ (or $d_2 = (t, p, n, l_2)$) and $l_1 > l_2$. Then $c(p, d_1) \leq c(p, d_2)$.*

Proof. Proof: Assume disruption d_1 becomes known at period t_1 and disruption d_2 becomes known at period t_2 for $t_1 < t_2$. The proof would be trivial if at the earlier time t_1, one could wait until the later time t_2 to take action. However, if rerouting is chosen in our recovery model, it needs to be done immediately at the time of recovery. Consider any ship rerouted in the recovery at time t_2. Rerouting the ship to the same new destination at time t_1 instead will yield a lower (or equal) cost given the triangle inequality, and will remain feasible because its arrival time at the new destination will be no later. Therefore, $c(p, d_1) \leq c(p, d_2)$. □

Proposition 2 can be used to obtain a lower bound on the actual cost for disruptions with the same location and intensity.

Corollary 1. *Assume $c(d^*)$ is the optimal cost of the planning model with the disruption $d^* = (w, n)$ already known at the start of the planning horizon. Then $c(d^*) \leq c(p, d)$, where p is any planning solution, $d = (w, n, l)$ (or $d = (t, p, n, l)$) and l is any lead time up to the start of the horizon.*

4.2 A recovery model

The recovery model is used to reoptimize over the remaining periods once a disruption occurs. By following a two-stage format of planning and recovery, we focus on a single disruption in this study. However, real situations may involve sequential disruptions while the original schedule is being executed. Therefore, it is reasonable to include robustness in the recovery model despite the fact that only a single disruption is considered for each scenario. To achieve this, the Lagrangian terms are kept and the strategy of slack reallocation is also applied in the recovery model. The recovery model also includes the proposed recourse options. We use the same notation in the recovery model as in the planning model, but they actually indicate different sets since we solve a problem with a shorter time horizon in the recovery stage.

$$\min \; g_2(x) + M' \sum_{j \in \mathcal{J}^{\mathcal{D}}} \sum_{k \in \mathcal{K}_j} s_{jk} \tag{20}$$

$$s.t. \qquad \sum_{a \in \mathcal{FS}'(n_0, v)} x_a = 1, \qquad \forall v \in \mathcal{V} \tag{21}$$

$$\sum_{v \in \mathcal{V}} \sum_{u_{jk} \leq t \leq v_{jk}} f_{nv} + s_{jk} = q_{jk}, \qquad \forall j \in \mathcal{J}^{\mathcal{D}}, \; \forall k \in \mathcal{K}_j \tag{22}$$

$$(2),\ (4)-(6),\ (8)-(10),\ (12).$$

The objective function (20) includes spot market costs where M' is a huge number ($M' > M^0$) so that the spot market is only used to avoid infeasibility. (21) is the flow conservation constraint on the source node, but with $\mathcal{FS}'(n_0, v)$ defined differently. Specifically, if the disruption becomes known at time $t-l$, then

$$\mathcal{FS}'(n_0, v) = \begin{cases} \{n_0 \to (j_0, 0)\} & \text{if ship } v \text{ is at port } j_0 \text{ at time } t-l, \\ \{n_0 \to (j, d_j)\ \forall j \in \mathcal{J}^{\mathcal{L}}\} & \text{if ship } v \text{ is on the route to some loading port at time } t-l, \\ \{n_0 \to (j, d_j)\ \forall j \in \mathcal{J}^{\mathcal{D}}\} & \text{if ship } v \text{ is on the route to some discharging port at time } t-l, \end{cases}$$

where d_j is the distance between the geographic location of ship v at time $t-l$ and the port j. (22) assures that demands are satisfied either from the ship deliveries or the spot market.

5 Computational results

The dual problem (17) incorporates two robustness strategies: the slack reallocation and the soft constraint approach. By running the simulator, we show their effects on solutions in terms of robustness. Section 5.1 gives a description of the instances we use and Section 5.2 shows the simulation results.

5.1 Test instances

A total of 18 instances are created based on the MIRPTW described in Section 2. The instances can be categorized into five classes according to the number of loading and discharging ports, and for each instance, a 60-period problem is defined. Depending upon different instances and ports, one-way voyage durations range from 3 to 9 periods. Table

1 provides some detailed information.

Class	C1	C2	C3	C4	C5
Instance No.	1-6	7-12	13,14	15,16	17,18
# of loading ports	2	1	2	2	1
# of discharging ports	2	3	3	3	4
# of ships	3	4	3	4	4
# of time windows	10-11	12-15	12-13	15	17-18

Table 1: Instances description

Once the simulator is called to evaluate a solution, three types of disruptions are generated in the simulation program:

1. Disruptions at loading ports. A disruption at a loading port is represented as a four dimensional vector (t, p, n, l). In the simulation program, we randomly generate five different pairs of (t, p), and then run a total of 30 disruptions that have the form $\{(t_i, p_i, n, l) : \; i = 1, \ldots, 5, \; n = 1, 2, 3, \; l = 2, 5\}$.

2. Disruptions of a single time window of a discharging port. A disruption at a discharging port is represented as a triple (w, n, l). In the simulation program, we enumerate the disruptions over all the time windows $\mathcal{K}$, and run a total of $6|\mathcal{K}|$ disruptions that have the form $\{(w_{jk}, n, l) : \; j \in \mathcal{J}^{\mathcal{D}}, \; k \in \mathcal{K}_j, \; n = 1, 2, 3, \; l = 2, 5\}$.

3. Disruptions of two time windows that are close in time. The purpose is to represent a big disruption that first hits one port and then another. Let $C = \{(w_{jk}, w_{j'k'}) : |u_{jk} - u_{j'k'}| \leq 5, \; j, j' \in \mathcal{J}^{\mathcal{D}}, \; k, k' \in \mathcal{K}_j, \; (j, k) \neq (j', k')\}$ be a set that defines all pairs of close time windows. In the simulation program, we randomly generate 6 min $\{5, |C|\}$ disruptions, each involving two close time windows with a $1, 2$ or 3-day disruption and a 2 or 5-day lead time.

We set the limit on the maximum number of iterations $\Omega = 10$ and the gap tolerance ϵ between the optimal value of $LR(\theta^k)$ and U as 1%. The integer programs are solved using CPLEX 12.5. The solver is stopped after 3600 seconds (planning model) or 600 seconds (recovery model). If no integer solution has been found, another 3600 or 600 seconds is given, repeating until an integer solution is found.

5.2 Simulation tests

Tables $2-4$ show the results of instances $1-18$. For every test instance, we select the solution from set A that has the lowest average simulated cost over all the scenarios and consider it as the robust solution. In each table, the second column gives the percent cost increase of the robust solution above the optimal planning solution, i.e.

$$\frac{\text{robust planning cost} - \text{original planning cost}}{\text{original planning cost}} * 100.$$

Let

$$AO = \frac{\text{average simulated cost of the original solution} - \text{original planning cost}}{\text{original planning cost}} * 100,$$

and

$$AR = \frac{\text{average simulated cost of the robust solution} - \text{robust planning cost}}{\text{robust planning cost}} * 100.$$

Then the first number in columns $3-8$ is AR, and the number in parenthesis is $AO-AR$.

The instances are grouped in three tables. For those in Table 2, the robust solution is not the optimal as planning solution. The Lagrangian approach produces a different solution with additional planned travel cost. However, the average simulated cost is reduced significantly when the robust plan is executed. In Figure 2, we provide scenario-by-scenario simulation results for Instance 4 as an example. The scenarios are sorted by the simulated cost improvement of the robust solution over the original solution. There were no instances where the planning solution cost increased without a significant performance improvement under disruption.

The contrast is even sharper in Table 3, where the robust plans are alternative optimal planning solutions. For some instances, the routing decision of the robust solution is different from that of the original plan, while for the others, their routing decisions are the same but with slacks allocated differently. In Figure 3, we show scenario-by-scenario simulation results for Instance 12. Without paying any price in the planning solution, we improve the flexibility of the system significantly.

However, our approach does not make much improvement for the instances shown in Table 4, and sometimes the robust solution is slightly inferior to the original solution. One possible reason is that given all the pre-specified hard time windows, it is likely that there exists no plan that could make all the deliveries flexible simultaneously. In this case, a planner has to make tradeoffs among these plans, or in other words, among various deliveries. We use Instance 17 to illustrate this. The scenario-by-scenario simulation results are given in Figure 4. The two solutions perform equally well under more than 85% of the scenarios, but neither dominates for all of the rest. Given the hard constraints, there is always some limitation on the flexibility of all the feasible solutions. When the original solution nearly reaches the flexibility limitation, our approach usually provides some alternative solutions that perform as well in an average sense, but might emphasize different deliveries in terms of flexibility. In addition, since the recovery model for reoptimizing the robust solutions includes the Lagrangian terms while the recovery model for reoptimizing the original solutions does not, it is likely that under some disruptions, the recovery costs associated with the robust solutions could be slightly higher than the original ones.

Finally, the lead time effect stated in Proposition 2 can be observed in the average simulated cost with 5 days of lead time as opposed to 2.

We hope that these results demonstrate that our approach is a promising way to deal with robustness and flexibility in maritime inventory routing problems and their applications.

Ins.	Planning cost increase	Average simulated cost over all scenarios $AR(AO - AR)$					
		lead time = 2			lead time = 5		
		1 day	1-2 days	1-3 days	1 day	1-2 days	1-3 days
1	10.2	0.0 (15.8)	0.0 (21.5)	4.4 (25.3)	0.2 (10.9)	0.2 (11.9)	4.6 (12.3)
3	1.7	11.1 (24.1)	22.5 (22.9)	29.9 (18.9)	3.9 (11.0)	7.6 (14.3)	13.4 (13.7)
4	3.4	0.0 (22.6)	5.0 (25.6)	6.7 (32.8)	0.0 (5.6)	3.5 (13.5)	5.3 (21.1)
6	1.6	6.9 (15.6)	6.9 (18.8)	15.5 (14.8)	6.9 (11.6)	6.9 (13.1)	12.0 (12.8)
13	4.8	4.4 (11.2)	7.0 (18.1)	10.9 (28.8)	4.4 (10.5)	7.1 (11.9)	10.2 (19.6)
14	5.4	13.5 (35.4)	14.2 (35.8)	20.0 (34.8)	10.5 (6.5)	11.4 (10.2)	17.3 (7.6)
15	1.2	4.4 (17.4)	8.8 (23.2)	15.1 (23.2)	1.4 (13.8)	5.3 (16.8)	11.1 (16.9)

Table 2: Simulation results for instances where the robust solution increases the planning cost but performs significantly better when disrupted.

Ins.	Planning cost increase	Average simulated cost over all scenarios $AR(AO - AR)$					
		lead time = 2			lead time = 5		
		1 day	1-2 days	1-3 days	1 day	1-2 days	1-3 days
2	0.0	12.7 (24.7)	27.6 (16.6)	36.9 (12.7)	11.4 (11.6)	21.6 (9.0)	31.3 (6.7)
5	0.0	9.9 (17.5)	12.9 (30.3)	17.3 (31.3)	6.3 (0.5)	9.5 (6.5)	14.3 (12.3)
8	0.0	7.4 (14.5)	9.6 (15.7)	14.6 (15.6)	5.4 (8.8)	7.0 (13.3)	11.8 (12.7)
9	0.0	5.5 (5.7)	7.4 (8.4)	10.1 (14.3)	4.2 (3.1)	6.0 (3.1)	7.0 (7.2)
11	0.0	8.4 (6.0)	11.4 (5.7)	15.1 (9.1)	8.3 (0.4)	10.1 (2.0)	11.9 (7.1)
12	0.0	0.0 (14.0)	3.0 (14.5)	9.2 (17.9)	0.0 (7.9)	1.9 (8.8)	8.4 (12.5)

Table 3: Simulation results for instances where the robust solution performs significantly better than the original planning solution, and with no increase in planning cost.

Ins.	Planning cost increase	Average simulated cost over all scenarios $AR(AO - AR)$					
		lead time = 2			lead time = 5		
		1 day	1-2 days	1-3 days	1 day	1-2 days	1-3 days
7	0.0	5.7 (0.0)	7.9 (-0.1)	10.4 (0.0)	1.0 (0.0)	5.3 (0.0)	8.5 (-0.1)
10	0.0	0.0 (0.0)	1.6 (0.0)	7.0 (0.3)	0.0 (0.0)	1.5 (0.0)	6.1 (-1.2)
16	0.0	4.1 (-0.4)	4.1 (-0.4)	10.4 (-0.5)	0.9 (-0.4)	1.3 (-0.5)	6.3 (-0.5)
17	0.0	1.7 (0.0)	4.4 (-0.1)	9.2 (-0.2)	0.5 (0.0)	3.0 (0.0)	7.8 (0.4)
18	0.0	4.9 (0.0)	13.0 (-0.2)	18.6 (-0.1)	1.5 (0.0)	8.8 (-0.1)	14.4 (0.2)

Table 4: Simulation results for instances with no increase in planning costs and little difference in performance when disrupted.

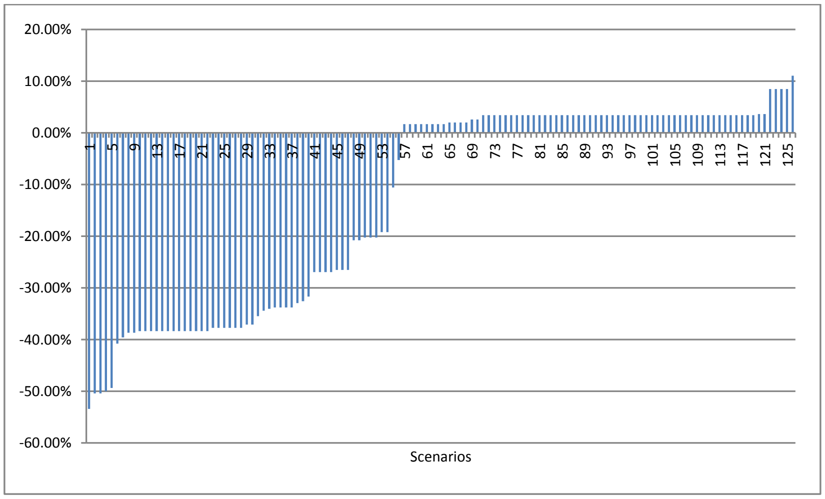

Figure 2: Simulated cost difference (robust/original) for Instance 4

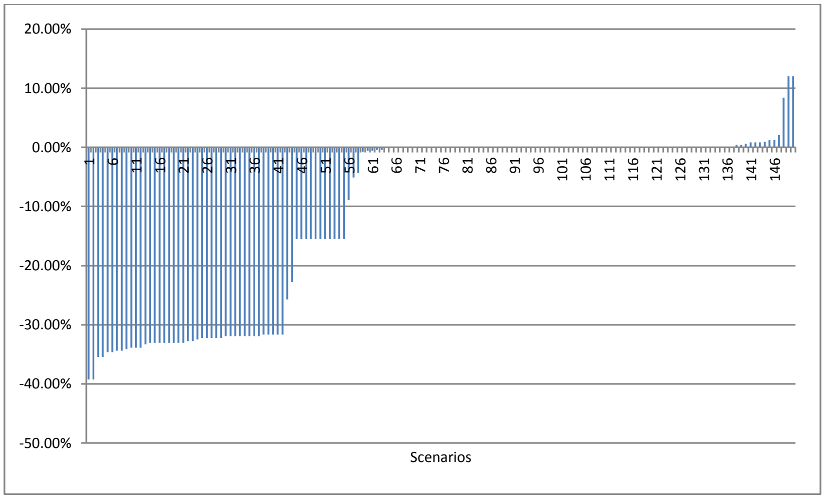

Figure 3: Simulated cost difference (robust/original) for Instance 12

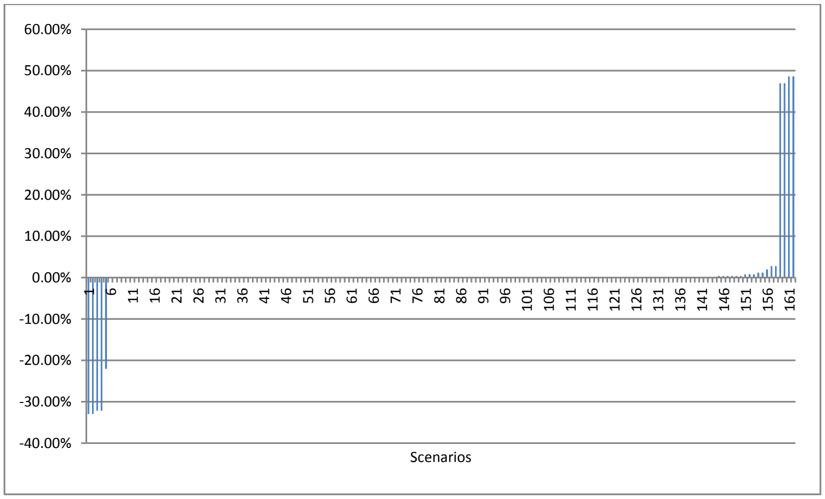

Figure 4: Simulated cost difference (robust/original) for Instance 17

Acknowledgments

We wish to thank Dimitri Papageorgiou of ExxonMobil Research and Engineering Company for his valuable suggestions, and thank ExxonMobil Research and Engineering Company for financial support of this research.

References

Ageeva, Y. 2000. Approaches to incorporating robustness into airline scheduling. Ph.D. thesis, Massachusetts Institute of Technology.

Agra, A., M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, C. Requejo. 2012. Layered formulation for the robust vehicle routing problem with time windows. *Combinatorial Optimization, Second International Symposium, ISCO 2012, Athens, Greece, April 19-21, 2012, Revised Selected Papers*. Springer, 249–260.

Agra, A., M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, C. Requejo. 2013. The robust vehicle routing problem with time windows. *Computers & Operations Research* **40** 856–866.

Alvarez, J. F., P. Tsilingiris, E. S. Engebrethsen, N. MP Kakalis. 2011. Robust fleet sizing and deployment for industrial and independent bulk ocean shipping companies. *INFOR: Information Systems and Operational Research* **49** 93–107.

Ben-Tal, A., L. El Ghaoui, A. Nemirovski. 2009. *Robust optimization*. Princeton University Press.

Cacchiani, V., A. Caprara, M. Fischetti. 2012. A lagrangian heuristic for robustness, with an application to train timetabling. *Transportation Science* **46** 124–133.

Chiraphadhanakul, V. 2010. Routing and scheduling models for robust allocation of slack. Ph.D. thesis, Massachusetts Institute of Technology.

Christiansen, M., K. Fagerholt. 2002. Robust ship scheduling with multiple time windows. *Naval Research Logistics* **49** 611–625.

Christiansen, M., K. Fagerholt, B. Nygreen, D. Ronen. 2007. Maritime transportation. *Handbooks in Operations Research and Management Science*. Elsevier, 189–284.

Christiansen, M., K. Fagerholt, D. Ronen. 2004. Ship routing and scheduling: Status and perspectives. *Transportation Science* **38** 1–18.

Fischetti, Matteo, Michele Monaci. 2009. Light robustness. *Robust and online large-scale optimization*. Springer, 61–84.

Halvorsen-Weare, E. E., K. Fagerholt, M. Rönnqvist. 2013. Vessel routing and scheduling under uncertainty in the liquefied natural gas business. *Computers & Industrial Engineering* **64** 290–301.

Held, M., P. Wolfe, H. P. Crowder. 1974. Validation of subgradient optimization. *Mathematical programming* **6** 62–88.

Lan, S., J.-P. Clarke, C. Barnhart. 2006. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science* **40** 15–28.

Papageorgiou, D. J., M.-S. Cheon, A. B. Keha, G. L. Nemhauser, J. Sokol. 2012. Mirplib: A maritime inventory routing survey and instance library. Georgia Institute of Technology.

Rosenberger, J. M., E. L. Johnson, G. L. Nemhauser. 2003. Rerouting aircraft for airline recovery. *Transportation Science* **37** 408–421.

Rosenberger, J. M., E. L. Johnson, G. L. Nemhauser. 2004. A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science* **38** 357–368.

Schaefer, A. J., E. L. Johnson, A. J. Kleywegt, G. L. Nemhauser. 2005. Airline crew scheduling under uncertainty. *Transportation Science* **39** 340–348.

Shapiro, A., D. Dentcheva, A. Ruszczyński. 2009. *Lectures on stochastic programming: modeling and theory*, vol. 9. Society for Industrial and Applied Mathematics.

Shebalov, S., D. Klabjan. 2006. Robust airline crew pairing: Move-up crews. *Transportation Science* **40** 300–312.

Smith, B. C., E. L. Johnson. 2006. Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science* **40** 497–516.

Song, J.-H., K. C. Furman. 2013. A maritime inventory routing problem: Practical approach. *Computers & Operations Research* **40** 657–665.

Yen, J. W., J. R. Birge. 2006. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science* **40** 3–14.