

Robust Inventory Routing with Flexible Time Window Allocation

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 Dimitri Papageorgiou

Corporate Strategic Research

ExxonMobil Research and Engineering Company, Annandale, New Jersey 08801

myun-seok.cheon@exxonmobil.com dimitri.j.papageorgiou@exxonmobil.com

This paper studies a robust maritime inventory routing problem with time windows and stochastic travel times. One of the novelties of the problem is that the length and placement of the time windows are also decision variables. Such problems arise in the design and negotiation of long-term delivery contracts with customers who require on-time deliveries of high-value goods throughout the year. We formulate the problem as a two-stage stochastic mixed-integer program and propose a two-phase solution approach that considers a sample set of disruptions as well as their recovery solutions. In the first phase, we introduce two planning strategies to generate robust routes in which time buffers are spread among deliveries and consecutive deliveries at a port are separated by at least some minimum number of periods. In the second phase, we propose a multi-scenario construction heuristic to obtain good feasible solutions. Computational results reveal that our integrated solution procedure with judicious placement of time buffers and committed time windows leads to robust solutions that are less vulnerable to unplanned disruptions and have lower expected costs.

Key words: Inventory routing, Robust planning, Time windows, Stochastic programming, Construction heuristic

1 Introduction

We study a robust maritime inventory routing problem with time windows for deliveries with uncertain disruptions, where the length and placement of the time windows are also decision variables. In a traditional inventory routing problem, a vendor is responsible for both the inventory management at suppliers and customers, and for the routing of vehicles to pick up and deliver products. A variant is to assume that time windows are present (as given data) to account for the fact that pick-ups and deliveries may only be permitted within pre-specified time intervals. In this paper, we consider the problem faced by a single vendor who must simultaneously decide routes for all the ships and delivery time windows.

The motivation for this work stems from vendor managed inventory problems where a vendor

is responsible for delivering product to several customers over a planning horizon. Because the customers receive product from numerous sources, they typically negotiate delivery amounts and delivery times with each of their vendors. Customers do not reveal their actual inventory levels or consumption rates to the suppliers. Instead, they enter contractual agreements with vendors for deliveries over the planning horizon. Meanwhile, from the vantage point of a single vendor, the problem becomes an inventory routing problem with time windows (IRPTW) in which the time windows are decision variables. The vendor must provide to each individual customer a list of time windows for the entire planning horizon. There is an important tradeoff to be made when generating these time windows. On the one hand, customers prefer to have small time windows to better plan their day-to-day operations, reduce their inventory levels, and lower their overall risk exposure. On the other hand, the vendor prefers to have large time windows so that it is possible to meet all contractual requirements even in the presence of disruptions in the planned delivery routes.

There are two fundamental ways to better withstand disruptions. First, the vendor can strategically develop routes possessing characteristics that allow for flexible recovery when a disruption occurs. Second, he can judiciously place delivery time windows at customers so that there are more opportunities to satisfy a delivery. However, in the absence of disruptions, both options may incur an additional cost above the optimal deterministic delivery solution. Since the vendor is interested in developing a delivery plan that is both economical as well as robust against uncertain disruptions, it is more favorable to have an integrated solution procedure that simultaneously considers the routing and time window allocation. Therefore, the objective of the robust inventory routing problem with time window allocation considered in this paper is to determine robust routes for all the vehicles and flexible time windows at all the customers under unknown disruptions, so as to minimize the total expected costs.

An important motivating application of this work is in the creation and negotiation of an Annual Delivery Plan (ADP) in the liquefied natural gas (LNG) business. The overall ADP planning activity is to develop contractual agreements of delivery plans that specify delivery dates (or time windows) and the corresponding delivery quantities. We refer to Grønhaug and Christiansen (2009) and Andersson et al. (2010a) for a review of the LNG supply chain. Grønhaug et al. (2010) propose a branch-and-price method and implement different accelerating strategies to solve a LNG inventory routing problem. Rakke et al. (2011), Stålhane et al. (2012) and Rakke et al. (2014) study a deterministic ADP problem in the LNG supply chain. The goal is to find an optimal plan for a heterogeneous fleet of ships that delivers two types of products from a single depot to a set of discharging ports such that constraints related to inventory storage at loading ports and contractual obligations at discharging ports are satisfied. Delivery time windows are not considered in their studies.

1.1 Relevant studies

Inventory routing problems (IRPs) involve a combination of inventory management and vehicle routing. Andersson et al. (2010b) and Coelho et al. (2013) give comprehensive reviews of IRPs. “Several applications of the IRP have been documented. Most arise in maritime logistics, namely in ship routing and inventory management.... Problems arising in the chemical components industry and in the oil and gas industries are also a frequent source of applications in a maritime environment” (Coelho et al. (2013), p.2). Surveys on maritime inventory routing problems are given by Christiansen et al. (2013) and Papageorgiou et al. (2014b).

The IRPTW is a variant of the IRP in which time windows for pick-ups and deliveries are given exogenously (as data). Early work in the maritime sector includes Christiansen and Nygreen (1998) and Christiansen (1999) who study the inventory pickup and delivery problem with time windows, which involves scheduling of a fleet of ships while respecting the service time windows and inventory restrictions. Fagerholt (2001) studies a multi-ship pickup and delivery problem with soft time windows. The objective is to find shipping schedules with significant reduction in transportation cost by introducing soft time windows which can be violated at the expense of paying inconvenience cost. For a similar problem, Christiansen and Fagerholt (2002) 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. Agra et al. (2012) and Agra et al. (2013) investigate a vehicle routing problem with time windows (VRPTW) 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. Zhang et al. (2013) develop a Lagrangian heuristic scheme for generating robust solutions to a maritime inventory routing problem with time windows with uncertain travel disruptions.

However, the problem of assigning time windows has largely been overlooked in the VRP and IRP literature. To the best of our knowledge, this is the first study to consider an inventory routing problem with time window allocation. Spliet and Gabor (2014) consider a related problem, referred to as the time window assignment vehicle routing problem, which focuses exclusively on demand uncertainty. It consists of determining a time window assignment before demand is known, and finding a vehicle routing schedule for each scenario satisfying the time windows such that the expected costs are minimized. Another related work is the vehicle routing problem with self-imposed time windows (VRP-SITW) studied in Jabali et al. (2013). Unlike in the traditional VRPTW where time windows are given exogenously by the customers, in the VRP-SITW, the vendor assigns customers to vehicles, sequences the customers allocated to each vehicle, and sets the time windows in which it plans to serve the customers. The term “self-imposed” refers to the fact that the vendor selects the time windows by itself, independently of the customer, and therefore time windows are treated as endogenous to the routing problem. The work also aims to cope with travel time disruptions by inserting slack time into the schedule. Like the VRP-SITW, our problem treats time window allocation as a decision variable to cope with various disruptions. However, compared with the VRP-SITW, we consider a more complex problem which involves a heterogeneous fleet of ships, a multi-period planning horizon and multiple loading ports. Also, unlike in the VRP-SITW where the length of time windows is given, we treat it as a decision variable in our problem.

The maritime industry is directly impacted by a variety of disruptions. Christiansen et al. (2006) discuss some problems from the shipping industry where robustness plays an important role and categorize them into strategic, tactical and operational planning problems. At the strategic level, the uncertainties can affect the quality of decisions regarding fleet sizing and composition. At the tactical level, they state that “several unpredictable factors influence the fulfillment of plans and should be considered in the planning process. The two most important are probably: (1) weather conditions that can strongly influence the sailing time, and (2) port conditions such as strikes and mechanical problems that can affect the time in port” (p.273). At the operational level, we may consider delays due to tides and restricted opening hours at ports. In this paper, we implicitly assume that minor disruptions, i.e., those that can be recovered from by actions such as speeding up ships, are accounted for operationally and thus are not considered in our model.

Instead, we focus on major disruptions, including work stoppages and severe weather conditions, that may result in several days of delay on all routes at a specific port or in a region.

There are only a few studies that deal with robust planning in the shipping industry. Besides what is mentioned above in Christiansen and Fagerholt (2002), Agra et al. (2012), Agra et al. (2013) and Zhang et al. (2013), a 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 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. Tirado et al. (2013) apply three heuristics to a dynamic and stochastic maritime routing problem, and demonstrate that average cost savings of 2.5% can be achieved by including stochastic information in the model.

1.2 Contributions

The main contributions of this paper are:

1. We introduce a robust maritime inventory routing problem with time window allocation (MIRPTWA) where the length and placement of the time windows are also decision variables. The problem involves a combination of not only inventory management and vehicle routing, but also time window placement. To the best of our knowledge, this generalized maritime inventory routing problem has not been studied in the literature.
2. We formulate the problem as a two-stage stochastic mixed-integer program and propose a two-phase solution approach that considers various types of disruptions that may affect the travel time or availability of vessels to deliver in certain time periods.
3. We develop modeling techniques that aim to generate robust routes in which time buffers are spread among deliveries and consecutive deliveries at a port are separated by at least a minimum number of periods.
4. We propose a stochastic multi-scenario construction heuristic to reduce the solution times in determining good and flexible time windows.

The remainder of the paper is organized as follows. Section 2 provides a description of the problem and discusses the types of disruptions considered in this paper. Section 3 gives the mathematical model formulation and proposes a two-phase solution approach. Section 4 presents two robustness strategies for generating robust routes, and gives the Phase I computational results. Section 5 proposes a stochastic multi-scenario construction heuristic, and solves a two-stage stochastic program with recourse that considers a set of disruptions and their recovery solutions to determine delivery time windows. We also report the Phase II computational results. In Section 6, we give concluding remarks.

2 Problem Description

The MIRPTWA is defined on a finite planning horizon where set \mathcal{T} contains all time periods. We propose a two-stage stochastic programming model by considering a set \mathcal{S} of disruption scenarios where scenario 0 has no disruptions and each scenario $s \in \mathcal{S}/\{0\}$ contains exactly one disruption. The solution under scenario 0 is called the *original plan*. For each disruption scenario, the planning

horizon is partitioned into two segments. In the first segment, the original plan is executed; in the second, a *recovery plan* is implemented after period l_s , $s \in \mathcal{S}/\{0\}$ when the disruption is known. The disruption might not occur until a time after l_s , for example in the case of an approaching hurricane. In other words, we consider the effect of lead time. In reality, multiple disruptions might occur when an original plan is executed. We approximate the true problem by solving a two-stage stochastic programming model recursively. By using the same planning model at the recovery procedure, it can be viewed as a re-planning stage for potential future disruptions.

We assume there is a set of loading ports denoted by $\mathcal{J}^{\mathcal{L}}$. The i -th port has a constant production rate $p_{i,t}$ of a single product at period $t \in \mathcal{T}$ and an initial inventory level $I_{i,0}$. The variable $I_{i,t}^s$ tracks the inventory level at port $i \in \mathcal{J}^{\mathcal{L}}$ at the end of period $t \in \mathcal{T}$ under scenario $s \in \mathcal{S}$, and it must be between a lower limit L_i and an upper limit U_i in each period.

We consider a set of discharging ports denoted by $\mathcal{J}^{\mathcal{D}}$. Contractual agreements (made prior to the planning decisions considered here) stipulate that the vendor must deliver at least quantity Q_j^{Port} of product at discharging port $j \in \mathcal{J}^{\mathcal{D}}$ over the entire planning horizon. We use a heterogeneous fleet of ships denoted by $\mathcal{V} = \{1, 2, \dots, V\}$, with each ship $v \in \mathcal{V}$ having some load capacity Q_v^{Ship} to deliver the product. For simplicity, we restrict our attention to problems where ships always fully load or discharge at a port, and assume that ships leave a port immediately after they load or discharge the product. The agreements also stipulate that the product should be delivered fairly evenly throughout the planning horizon. Thus, at each discharging port $j \in \mathcal{J}^{\mathcal{D}}$, we define a set \mathcal{K}_j of *time intervals* which corresponds to subsets of consecutive periods, and specify a targeted delivery quantity q_{jk} for each time interval $k \in \mathcal{K}_j$. These targets do not have to be met exactly, but there is a penalty cost $P^{\mathcal{D}}$ per unit associated with the deviation from the targets. We assume set \mathcal{T}_{jk} contains all the periods in time interval $k \in \mathcal{K}_j$ at discharging port $j \in \mathcal{J}^{\mathcal{D}}$. The time intervals at a discharging port might overlap with each other. In addition, the berth limit B_j is the maximum number of ships that can discharge in a given time period at discharging port $j \in \mathcal{J}^{\mathcal{D}}$. We assume that service time (time to load/discharge) is already built into the travel time. Let $\mathcal{J} = \mathcal{J}^{\mathcal{L}} \cup \mathcal{J}^{\mathcal{D}}$ be the set of all ports.

The planning problem determines a list of time windows at each discharging port. There is an upfront cost C_{jw}^{TW} if a w -day time window is placed at discharging port $j \in \mathcal{J}^{\mathcal{D}}$. We assume that the length of each time window is at most W periods and any two consecutive time windows at a discharging port are separated by at least Z periods. The time windows are not vessel-specific, but are the same across all the scenarios. A penalty cost $P^{\mathcal{O}}$ per unit incurs under scenario $s \in \mathcal{S}/\{0\}$ for each delivery its recovery plan requires outside of the time windows.

We assume that each scenario $s \in \mathcal{S}/\{0\}$ only includes a single disruption. While multiple disruptions could occur, we do not consider that case in this study. Two types of disruptions are considered in the paper:

- *Travel disruptions*, such as those caused by a hurricane, affect all routes of ships inbound to a port, resulting in one or multiple day(s) of delay on all routes scheduled to enter the port in a certain time interval.
- *Port disruptions*, such as those caused by a strike or maintenance issue at a specific port, result in one or multiple day(s) of delay for all ships scheduled to load or discharge at the port during a certain time interval.

Mathematically, the two types of disruptions can be defined as follows. Assume $D(s) = \{(i, t, d)\}$ is the disruption considered in scenario $s \in \mathcal{S}/\{0\}$. If it is a travel disruption, then all the ships

that are scheduled to arrive at port $i \in \mathcal{J}$ during periods that are close to $t \in \mathcal{T}$ are delayed by d days. If it is a port disruption, then all the ships at the port $i \in \mathcal{J}$ are not allowed to load or discharge during periods $t, \dots, t + d - 1$. Since we assume that ships leave the ports immediately after they load or discharge the product, equivalently, one can interpret the disruptions as (travel disruption): all the ships that are scheduled to arrive at port i during periods that are close to $t \in \mathcal{T}$ are forced to stay at the port by at least d days before departure, and (port disruption): ships are not allowed to leave port i during periods $t, \dots, t + d - 1$.

The MIRPTWA consists of determining routes for all the ships (including original plan and recovery plan under each scenario $s \in \mathcal{S}/\{0\}$), as well as time windows at all the discharging ports so as to minimize total expected costs of travel, placing delivery time windows, over/under-deliveries, and deliveries outside of the time windows over all the scenarios. We assume equal probability for each scenario $s \in \mathcal{S}$.

3 Mathematical Formulations

In this section, we first present a two-stage stochastic mixed-integer programming model as a Full Formulation (FF) for the MIRPTWA and then propose a two-phase heuristic approach to solve the problem.

3.1 Full Model

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 waiting 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 waiting arcs from node (j, t) to node $(j, t + 1)$ represent staying at port j in both period t and $t + 1$. We assume the travel arcs include the time to load or discharge. We use \mathcal{A} to denote the set of all arcs, and \mathcal{A}^+ to denote the set of all travel arcs. We assume C_a^T is the travel cost associated with arc $a \in \mathcal{A}^+$. There is no cost for keeping a ship at a port. 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. Let $y_{jtw} = 1$ if there exists a w -day time window ($0 \leq w \leq W$) starting from period $t \in \mathcal{T}$ at discharging port $j \in \mathcal{J}^D$, and $y_{jtk} = 0$ otherwise. The decision variable f_{jk}^s represents the over-delivery quantity of time interval $k \in \mathcal{K}_j$ at discharging port $j \in \mathcal{J}^D$ under scenario $s \in \mathcal{S}$. If period $t \in \mathcal{T}$ at discharging port $j \in \mathcal{J}^D$ is not covered by any time window, the decision variable g_{jt}^s represents the delivery quantity under scenario s ; otherwise, $g_{jt}^s = 0$. An arc-flow mixed-integer programming model is given by:

$$(\mathbf{FF}) \quad \min \sum_{a \in \mathcal{A}^+} C_a^T x_a^0 + \sum_{j \in \mathcal{J}^D} \sum_{t \in \mathcal{T}} \sum_{w=1}^W C_{jw}^{\text{TW}} y_{jtw} + P^D \sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} f_{jk} + P^O \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}^D} \sum_{t \in \mathcal{T}} g_{jt}^s \quad (1)$$

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

$$\sum_{a \in \mathcal{FS}(n_0, v)} x_a^s = 1, \quad \forall s \in \mathcal{S}, \forall v \in \mathcal{V}. \quad (3)$$

$$\sum_{a \in \mathcal{RS}(n_T, v)} x_a^s = 1, \quad \forall s \in \mathcal{S}, \forall v \in \mathcal{V}. \quad (4)$$

$$I_{i,t}^s = I_{i,t-1}^s + p_{i,t} - \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in \mathcal{FS}(i,t,v)^+} x_a^s, \quad \forall s \in \mathcal{S}, \forall i \in \mathcal{J}^{\mathcal{L}}, \forall t \in \mathcal{T}. \quad (5)$$

$$L_i \leq I_{i,t}^s \leq U_i, \quad \forall s \in \mathcal{S}, \forall i \in \mathcal{J}^{\mathcal{L}}, \forall t \in \mathcal{T}. \quad (6)$$

$$\sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^s \leq B_j, \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J}^{\mathcal{D}}, \forall t \in \mathcal{T}. \quad (7)$$

$$\sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^s \geq Q_j^{\text{Port}}, \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J}^{\mathcal{D}}. \quad (8)$$

$$\sum_{t \in \mathcal{T}_{jk}} \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^0 - f_{jk} \leq q_{jk}, \quad \forall j \in \mathcal{J}^{\mathcal{D}}, \forall k \in \mathcal{K}_j. \quad (9)$$

$$x_{(i,u,v) \rightarrow (i,u+1,v)}^s \geq \sum_{a \in \mathcal{RS}(i,t,v)^+} x_a^s, \quad u = t, \dots, t+d-1, (i,t,d) \in D(s), \forall s \in \mathcal{S}/\{0\}, \forall v \in \mathcal{V}. \quad (10)$$

$$\sum_{u=t}^{t+d-1} \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(i,t,v)^+} x_a^s = 0, \quad (i,t,d) \in D(s), \forall s \in \mathcal{S}/\{0\}, \forall v \in \mathcal{V}. \quad (10')$$

$$x_a^s = x_a^0, \text{ if arc } a \text{ is a decision to be made before } l_s, \quad \forall s \in \mathcal{S}/\{0\}. \quad (11)$$

$$\sum_{w=1}^W y_{jtw} \leq 1, \quad \forall j \in \mathcal{J}^{\mathcal{D}}, \forall t \in \mathcal{T}. \quad (12)$$

$$y_{jtw} + y_{jt'w'} \leq 1, \quad \forall j \in \mathcal{J}^{\mathcal{D}}, t < t', t+w+Z+1 \geq t'. \quad (13)$$

$$g_{jt}^s \geq \sum_{v \in \mathcal{V}} Q_v^{\text{Ship}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^s - M \sum_{\tau=t-W+1}^t \sum_{w=t+1-\tau}^W y_{i\tau w}, \quad \forall s \in \mathcal{S}, \forall j \in \mathcal{J}^{\mathcal{D}}, \forall t \in \mathcal{T}. \quad (14)$$

$$x_a^s, y_{jtw} \in \{0,1\}, f_{jt} \geq 0, g_{jt}^s \geq 0, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \forall j \in \mathcal{J}^{\mathcal{D}}, \forall t \in \mathcal{T}, w = 1, \dots, W. \quad (15)$$

The objective is to minimize total costs of travel, placing delivery time windows, over/under-deliveries in all the considered time intervals, and deliveries outside of the time windows over all the scenarios. Network flow constraints (2) – (4) require each ship to travel from the source to the sink in each scenario. (5) and (6) are balance constraints of the product at loading ports. (7) are

the berth constraints at discharging ports. Constraints (8) and (9) ensure that a minimal quantity of the product is delivered at each discharging port and it is delivered fairly evenly throughout the planning horizon. Constraints (10) and (10') correspond to the travel and port disruptions respectively. Nonanticipativity constraints (11) ensure that the original plan under scenario 0 is executed until period l_s when the disruption is realized in scenario $s \in \mathcal{S}/\{0\}$. Logical constraints (12) and (13) restrict the selection of time windows. Constraints (14) track the deliveries outside of the time windows, where M is a large number. Constraints (15) are the variable restrictions.

3.2 Decomposition Models for a Two-phase Heuristic Approach

Given the computational complexity, we are not able to consider all possible scenarios or all possible realizations of constraints (10) or (10') in a single model. However, most of the scenarios would not affect the original schedule if they disrupt a route on which there is no ship or a port where no loading/discharging is happening at the time.

As a remedy, we propose a two-phase heuristic approach to solve the problem. In Phase I, we generate routes by incorporating some robustness strategies into a deterministic inventory routing problem without considering disruption scenarios. In Phase II, given the routes obtained in Phase I, we determine delivery time windows by solving a restricted version of Model (FF) by including *heuristic constraints* that are used to fix the ships' visit sequences and a selected set of disruptions that may affect the given routes. When $\mathcal{S} = \{0\}$, i.e., when no disruptions are considered, the base model in Phase I is given as:

$$\begin{aligned} (\mathbf{P1F}) \quad & \min \sum_{a \in \mathcal{A}^+} C_a^T x_a^0 + P^D \sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} f_{jk} \\ & \text{s.t. (2) - (9),} \end{aligned} \quad (16)$$

$$x_a^0 \in \{0, 1\}, \quad f_{jt} \geq 0, \quad \forall a \in \mathcal{A}, \forall j \in \mathcal{J}^D, \forall t \in \mathcal{T}. \quad (17)$$

Based on (P1F), we introduce two planning strategies to generate robust routes in Phase I, which will be discussed in detail in Section 4. The resulting solutions of the Phase I model are routes for each ship that consist of visit sequences as well as loading and discharging times.

Given the solutions to the Phase I model, we extract the information regarding the visit sequences from the solutions and use it as an input in the Phase II model. Although the ships' visit sequences are given, the timing decisions may vary across all the scenarios. Let set \mathcal{H}^O (\mathcal{H}^I) contain all the triples (i, t, v) if ship v departs from (arrives at) port i during time interval $[t - e, t + e]$ in the Phase I solution, where e is a small integer. With these restrictions, the Phase II model with the heuristic constraints can be given as:

$$(\mathbf{P2F}) \quad \min \sum_{j \in \mathcal{J}^D} \sum_{t \in \mathcal{T}} \sum_{w=1}^W C_{jw}^{\text{TW}} y_{jtw} + P^D \sum_{j \in \mathcal{J}^D} \sum_{k \in \mathcal{K}_j} f_{jk} + P^O \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}^D} \sum_{t \in \mathcal{T}} g_{jt}^s \quad (18)$$

$$\text{s.t. (2) - (9), (10) or (10'), (11) - (15),}$$

$$x_a^s = 0, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{FS}(i, t, v)^+, (i, t, v) \notin \mathcal{H}^O, \quad (19)$$

$$x_a^s = 0, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{RS}(i, t, v)^+, (i, t, v) \notin \mathcal{H}^I. \quad (20)$$

Since the visit sequences are fixed in Phase II, we do not include the travel cost in objective function (18). Constraints (19) and (20) ensure that in each scenario, loading and discharging times may be changed from the Phase I solutions even if the visit sequences are given. Algorithm 1 provides an overview of our two-phase heuristic approach.

Algorithm 1: A Two-phase Heuristic Approach

Phase I Generate robust routes.

1.1 Solve a deterministic inventory routing problem with robustness strategies.

- Evenly allocate idle time (see Section 4.1).
- Separate deliveries with minimum time requirement (see Section 4.2).

1.2 Fix the visit sequence of each ship

Phase II Determine delivery time windows (see Section 5).

2.1 Apply a multi-scenario construction heuristic to obtain initial feasible solutions

2.2 Solve a two-stage stochastic program with recourse that considers a set of disruptions and their recovery solutions.

3.3 An Aggregate Model Based on Ship Class

In order to reduce the solution times in (P1F), we use an aggregate model in which ship data are aggregated by ship class so that individual ship schedules are not distinguished in solutions. Such an aggregation technique was used in Papageorgiou et al. (2014a) to obtain a coarse solution in their system model. To apply the aggregate model, we create a variant of the time-space network described in Section 3.1, in which the arc set \mathcal{A}_{vc} for each ship class $vc \in \mathcal{VC}$ is the union of the arc sets for all the ships in the ship class. Assume there are N_{vc} ships in the ship class vc , then we replace constraints (2) – (4) and (17) in (P1F) with the following ones in the aggregate model, which is referred to as **(P1FVC)**.

$$\sum_{a \in \mathcal{FS}(n, vc)} x_a^0 - \sum_{a \in \mathcal{RS}(n, vc)} x_a^0 = 0, \quad \forall vc \in \mathcal{VC}, \forall n \in \mathcal{N}. \quad (2')$$

$$\sum_{a \in \mathcal{FS}(n_0, vc)} x_a^0 = N_{vc}, \quad \forall vc \in \mathcal{VC}. \quad (3')$$

$$\sum_{a \in \mathcal{RS}(n_T, vc)} x_a^0 = N_{vc}, \quad \forall vc \in \mathcal{VC}. \quad (4')$$

$$x_a^0 \in \{0, 1, \dots, N_{vc}\}, \quad f_{jt} \geq 0, \quad \forall a \in \mathcal{A}_{vc}, \forall vc \in \mathcal{VC}, \forall j \in \mathcal{J}^D, \forall t \in \mathcal{T}. \quad (17')$$

A solution produced by (P1FVC) specifies routes for all ship classes, but not for individual ships. To assign a route for each ship, we perform a post-processing step by solving a trivial feasibility problem with constraints (2) – (4) and $\sum_{a \in \mathcal{A}_{a'}} x_a^0 = \bar{x}_{a'}^0, \forall a' \in \mathcal{A}_{vc}, \forall vc \in \mathcal{VC}$, where each set $\mathcal{A}_{a'}$ contains the arcs that are aggregated into the arc a' , and $\bar{x}_{a'}^0$ is the optimal value obtained by solving (P1FVC).

4 Phase I: Robust Routing

In this section, we introduce two robustness strategies to generate robust routes in Phase I: *slack reallocation* and *delivery separation*.

4.1 Slack (Idle Time) Reallocation

Since a ship might have more time than needed to get from one port to another, there can be some slack in planning solutions. We define *slack days* as the days when a ship is idle at a discharging port. From a robustness point of view, slack days can be regarded as a buffer to protect on-time deliveries. To avoid the cases where some deliveries are over-protected while others are very fragile, we propose a slack reallocation strategy in order to “evenly” allocate the slack among all the deliveries. By using the modeling technique shown in Figure 1, we can track the number of slack days associated with each delivery. In the given example, the normal travel time from loading port A to discharging port B is 3 days and the travel cost is c . We artificially include 4-day and 5-day travel arcs from loading to discharging ports on a standard time-space network, and associate them with slightly lower costs $c - \epsilon_1$ and $c - \epsilon_2$ respectively where $0 < \epsilon_1 < \epsilon_2$. Since using a 4-day travel arc is equivalent to using a normal 3-day travel day (starting from the same period) plus a waiting arc but has a lower cost, the optimization problem would prefer the former one. Therefore, for an optimal solution, we can obtain the number of deliveries that are protected by no slack days, 1 slack day and at least 2 slack days by counting the usage of 3-day (from loading to discharging ports), 4-day and 5-day travel arcs.

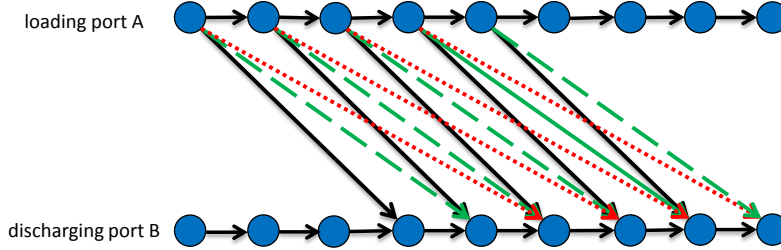


Figure 1: Slack reallocation

More generally, suppose a maximum of R slack days can give benefit as a buffer. To achieve the goal of “evenly” allocating the slack amongst all the deliveries, we assume that the difference between having a r -day slack and having a $(r + 1)$ -day is more significant than the difference between having a r' -day slack and having a $(r' + 1)$ -day if $r < r' \leq R$. Therefore, the discount factors of using artificial travel arcs are set as a concave function given in Figure 2.

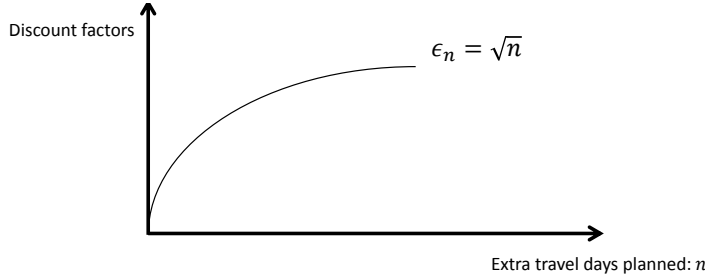


Figure 2: Discount factor for longer travel arcs

Let set \mathcal{A}_r^+ contain all the artificial travel arcs with an extra r travel days from loading to discharging ports ($r = 1, \dots, R$) and \mathcal{A}_0^+ contain all the normal travel arcs. We give a discount

for ships using artificial slow travel arcs; namely, $C_{a'}^T = C_a^T - \sqrt{r}$ if $a \in \mathcal{A}_0^+, a' \in \mathcal{A}_r^+$ and they represent the same port-to-port travel. To incorporate the slack reallocation strategy, we modified the objective function (16) as:

$$\min \sum_{r=0}^R \sum_{a \in \mathcal{A}_r^+} C_a^T x_a^0 + P^D \sum_{j \in \mathcal{J}^D} \sum_{k \in K_j} f_{jk}. \quad (16')$$

4.2 Separating Consecutive Deliveries

It is more likely that a disruption could cause significant problems when deliveries are clustered together during a short time frame at a port. Therefore, in order to mitigate the potential effect of a disruption, we separate any two consecutive deliveries at a port by a minimal number of periods. Suppose there are no more than one delivery within any consecutive U periods at a port, then we call U a separation parameter and rewrite constraints (7) as

$$\sum_{u=t}^{t+U-1} \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{FS}(j,t,v)^+} x_a^0 \leq 1, \quad \forall j \in \mathcal{J}^D, \forall t \in \mathcal{T}. \quad (7')$$

4.3 Phase I Computational Results

In this section, we show the effects of the two robustness strategies on generating ship routes. Time windows will be placed in Phase II. We expect to see that evenly distributing slack days among the deliveries and setting minimum separation times between consecutive deliveries at a port result in a modest increase in the total travel cost relative to ignoring such robustness strategies altogether. The benefits of coupling these robust routes with judiciously chosen time windows will be shown in Section 5.3.

A total of 10 instances are created based on the problem described in Section 2. In each instance, we define a 60 periods problem in which there are 2 – 4 loading ports, 2 – 8 discharging ports, 2 ship groups consisting of 3 – 8 ships. We include 1 extra day, 2 extra days and 3 extra days travel arcs while applying the slack reallocation strategy, namely $R = 3$, and we let the separation parameter U vary from 1 to 4 in the experiments. The integer programs are solved using CPLEX 12.5 with a 4-hour time limit.

Figure 3 shows the average travel cost of the solutions relative to base model. The vertical axis represents the percentage of cost increase based on the basic Phase I model (P1F), which optimistically assumes that there are no disruptions. The horizontal axis gives the separation parameter U defined in the second robustness strategy. The dotted and solid lines correspond to the models with and without the slack reallocation strategy, respectively. Since any feasible solution to a robust model is feasible to (P1F), we observe in Figure 3 that the average travel cost increases when the robustness strategies are applied. For any given separation parameter, this cost increase is roughly between 0.4% and 0.8% on average. Since the benchmark travel cost used here assumes that no disruptions occur, an average cost increase under 1% seems acceptable.

Figure 4 shows the slack distribution of the best solution found by various models. The horizontal axis represents the average number of deliveries over all the instances with various slack days, and the vertical axis represents different models where SR indicates slack reallocation and U is the separation parameter. We observe that when the slack reallocation strategy is applied, the number of deliveries that are protected by 1 slack day largely increases while the number of

deliveries associated with no slack and more than 2 slack days decreases. Therefore, it achieves our objective of “evenly” allocating the slack amongst all the deliveries.

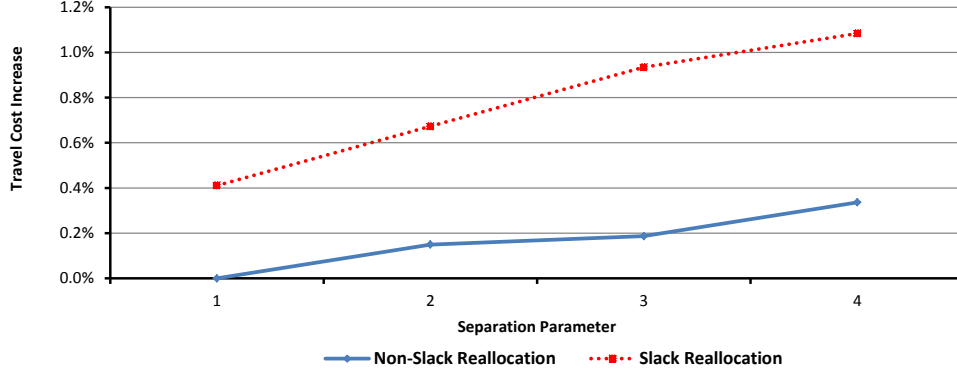


Figure 3: Phase I: Travel cost comparison relative to model without robustness strategy

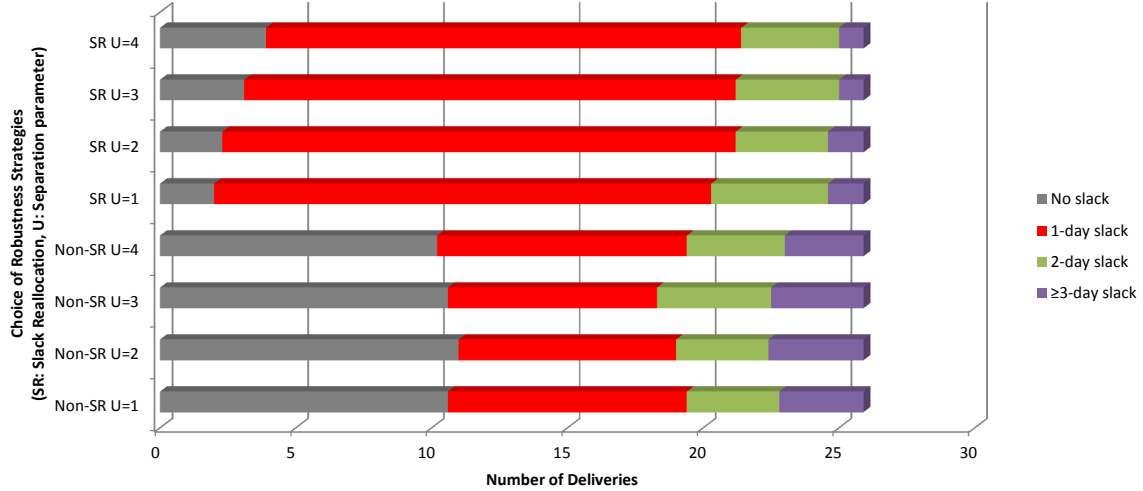


Figure 4: Phase I: Slack day comparison

5 Phase II: Time window placement

In Phase I, we obtain routes for ships that consist of both visit sequences and timing for loading and discharging. In Phase II, we allocate time windows for each delivery by solving a restricted version of Model (FF) in which the visit sequences of each ship, obtained in the Phase I solutions, are fixed.

5.1 Disruption Scenarios

Since the visit sequence is given for each ship while the loading and discharging times may vary across all the scenarios, constraints (19) and (20) force the value of a large set of decision variables to be 0 in the time-space network. In other words, we largely reduce the number of scenarios that might cause a disruption to the original schedule. To help understand how a disruption might affect travel, loading or discharging of a ship, we illustrate the two types of disruptions in Figure 5. The dashed lines represent original routes, and the solid lines represent recovery routes. In these two examples, a ship is affected due to a travel disruption (in the left one) and a port disruption (in

the right one) that both last for 2 periods. As a result, at least two deliveries are delayed due to the cascading effect. Although it shows no difference between these two disruptions in terms of their effects on this particular ship, they may cause different overall problems to the entire schedule. The travel disruption can delay some other inbound travel arcs from L1 to D1 as well that are close in period to the disrupted travel arc shown in the Figure. On the other hand, the port disruption can affect other ships that were prepared to discharge during the disrupted periods.

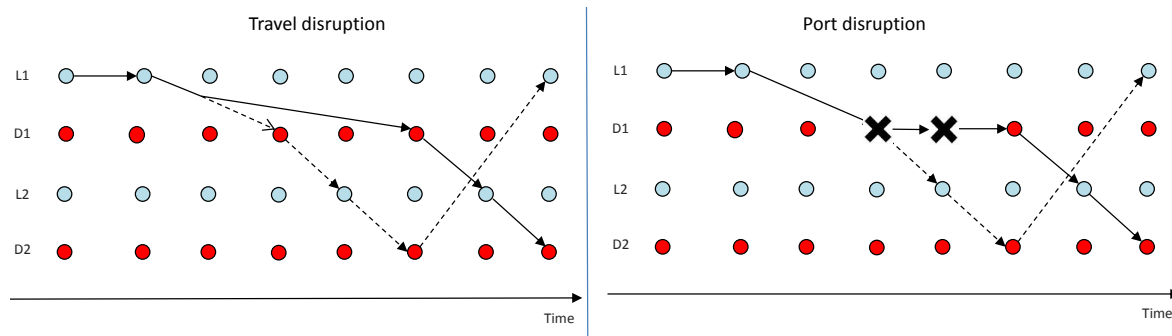


Figure 5: Types of disruptions and their impact on the network

5.2 Construction Heuristic

We propose a stochastic multi-scenario construction heuristic to obtain initial feasible solutions to (P2F). It can be viewed as an extension of the optimization-based local search heuristic that was first introduced in Song and Furman (2013). In their approach, in order to improve an initial feasible solution, they solve restricted optimization models. They fix the decision variables associated with all but two ships, optimize over those two ships, and repeat this procedure for all ship pairs. Goel et al. (2012) adapt this approach as a construction heuristic to generate solutions to instances with 365 time periods. In this paper, we propose a two-stage stochastic multi-scenario construction heuristic. The heuristic takes as input a set of feasible routes, one for each ship; that solution is obtained by solving a deterministic routing problem such as the one proposed by Zhang et al. (2013). In the first stage of the heuristic, we partition the ships into VG ship groups $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_{VG}$ based on how closely in time they visit the same port, and for each ship group, we solve a reduced optimization problem that considers only scenario 0 and every other scenario whose disruption may affect one or more ships in \mathcal{V}_{vg} , $vg \in \{1, 2, \dots, VG\}$. We merge all the individual ship-group solutions together to create an initial solution, which might not be feasible due to the inventory and berth constraints. Therefore, we proceed to a second stage similar to that of Song and Furman (2013), but generalized to allow sets of up to K ships. The second stage emphasizes eliminating feasibility until a feasible solution is obtained; once we have a feasible solution, we continue running improvement iterations in order to improve the quality of solutions. The output of the heuristic is the best solution found, which is then given to the full model as a starting solution for the MIP. An overview of the heuristic is given in Algorithm 2.

Algorithm 2: Stochastic Multi-scenario Construction Heuristic

Input: Set of routes for each ship (under no disruptions)

begin

Construction Stage

1. Partition the ships into VG ship groups.

1.1 For each ship pair $v_1, v_2 \in \mathcal{V}$, calculate $cl(v_1, v_2)$, which is the smallest difference in time between a loading/discharging of v_1 and a loading/discharging of v_2 at the same port.

1.2 Partition all the ships into VG ship groups such that $\sum_{vg=1}^{VG} \sum_{v_1, v_2 \in \mathcal{V}_{vg}} cl(v_1, v_2)$ is minimized. Each ship group has at least one ship.

2. **for** $vg = 1, 2, \dots, VG$ **do**

2.1 Create a set \mathcal{D}_{vg} that contains scenario 0 and the scenarios whose disruption may affect one or more ships in \mathcal{V}_{vg} .

2.2 Solve a reduced two-stage stochastic MIP over the scenario set \mathcal{D}_{vg} by only including the decision variables associated with the ships in \mathcal{V}_{vg} .

2.3 Obtain the solutions for the ships in \mathcal{V}_{vg} under the scenarios in \mathcal{D}_{vg} .

3 Merge the solutions.

for each pair (v, s) **where** $v \in \mathcal{V}_{vg} \subset \mathcal{V}$ **and** $s \in \mathcal{S}$ **do**

if $s \in \mathcal{D}_{vg}$ **then**

use the solution obtained in 2.3,

else

use the solution under no disruption.

Improvement Stage

4. **for** $k = 1, \dots, K$ **do**

for $i_1 = 1, \dots, V$ **do**

.....

for $i_k = i_{k-1} + 1, \dots, V$ **do**

4.1 Fix decision variables associated with each ship v , $v \notin \mathcal{VF} = \{i_1, \dots, i_k\}$.

4.2 Solve the restricted two-stage stochastic MIP with the remaining variables.

4.3 Update the current solution.

5.3 Phase II Computational Results

In Section 4.3, we observed that by incorporating the two robustness strategies in Phase I, there is a modest increase in the total travel cost relative to the basic model which is overly optimistic by assuming no disruptions. In this section, we show the benefits of generating such robust ship routes when we determine delivery time windows. We expect to see that despite a modest increase in the travel cost, applying the robustness strategies enables the vendor to commit relatively small time windows to its customers. Furthermore, compared with the modest increase in generating robust routes, we achieve much more savings by placing small delivery time windows and paying a lower expected recovery cost in case of potential disruptions.

We present the Phase II computational results of the 10 test instances described in Section 4.3. We extract the information regarding the visit sequences of ships from the solutions generated in Phase I, and fix them as heuristic constraints in (P2F). Since various planning strategies can lead to different visit sequences, we compare their performance based on the following metrics:

travel cost, time-window cost, and total cost of the original plan; and the expected total cost and worst-case-scenario total cost. The number of disruption scenarios considered in (P2F) depends on the number of deliveries in the original plan. Specifically, we include one disruption scenario for each delivery. In our experiments, the number of disruption scenarios varies from 14 to 41 among different instances. For each delivery at a discharging port, we create a disruption scenario with a 2-day delay for all ships on inbound routes to the port. The lead time is assumed to be 7 days, and the maximal length of a time window is 3.

Table 1 shows the comparisons of the solutions generated under two different planning strategies. For each instance, we generate two visit sequences in Phase I, and give them as an input to (P2F). In the first one (VS1), no robustness strategy is used, while in the second one (VS2), we apply the strategy of slack reallocation with separation parameter $U = 3$. The numbers in Table 1 show the percentage in cost change (where a negative value is a cost improvement) of VS2 solutions compared to VS1 solutions. For example, in instance 1, although there was a 0.66% cost increase associated with travel costs, the percentage of cost decrease related to time-window costs was 18.92% and the total scenario 0 cost decreased by 3.17% when using a robust solution over that of ignoring robustness issues. Moreover, the scenario-based expected total cost and the worst-case scenario total cost dropped by 4.12% and 7.11%, respectively, by incorporating the robustness strategies. Since we incorporate the slack reallocation and delivery separation constraints when generating VS2, scenario 0 travel costs cannot decrease. The important observation is that generating routing solutions judiciously with slack reallocation and delivery separation leads to lower time window costs, and decreases the total scenario 0 cost in each of the instances. Furthermore, by applying the two robustness strategies, we have lowered the expected total costs and the worst-case-scenario costs as well, in every instance.

Ins.	Original plan travel cost	Time-window cost	Original plan total cost	Scenario-based expected total cost	Worst-case scenario total cost
1	0.66%	-18.92%	-3.17%	-4.12%	-7.11%
2	3.82%	-22.81%	-3.27%	-4.22%	-3.10%
3	0.31%	-3.41%	-0.48%	-0.60%	-2.10%
4	1.51%	-9.80%	-0.32%	-0.34%	-0.31%
5	1.41%	-30.16%	-5.80%	-5.78%	-7.05%
6	0.31%	-18.56%	-4.07%	-4.18%	-4.91%
7	0.39%	-2.78%	-0.30%	-0.29%	-0.30%
8	1.14%	-33.33%	-6.74%	-7.68%	-6.90%
9	1.00%	-29.55%	-4.49%	-4.49%	-4.49%
10	0.00%	-12.28%	-1.92%	-2.12%	-2.16%

Table 1: Phase II: Cost comparisons for various visit sequences

Figure 6 shows the percentages of the time windows with different sizes when we use the VS2 solution as an input to (P2F) for each instance. We observe that by assuming a 2-day delay for each disruption in our experiments, roughly 60% – 80% of the time windows contain only 1 day.

We also conducted an experiment to compare our approach with a naive heuristic in which we increase every travel arc by 1 day and assign a 2-day time window to each delivery. In terms of the penalty costs due to the deliveries outside of the time windows, this naive heuristic performs similar to the benchmark strategy in which we use VS1 solutions as an input to (P2F). However, this naive heuristic yields a very high up-front cost of placing time windows which reveals that placing a 2-day time window for each delivery is too conservative. Consequently, our approach in which we use VS2 solutions as an input to (P2F) outperforms this naive heuristic by an even larger

margin in terms of the total cost. In summary, this comparison provides further evidence for the need to judiciously place time windows in order to balance costs with robustness.

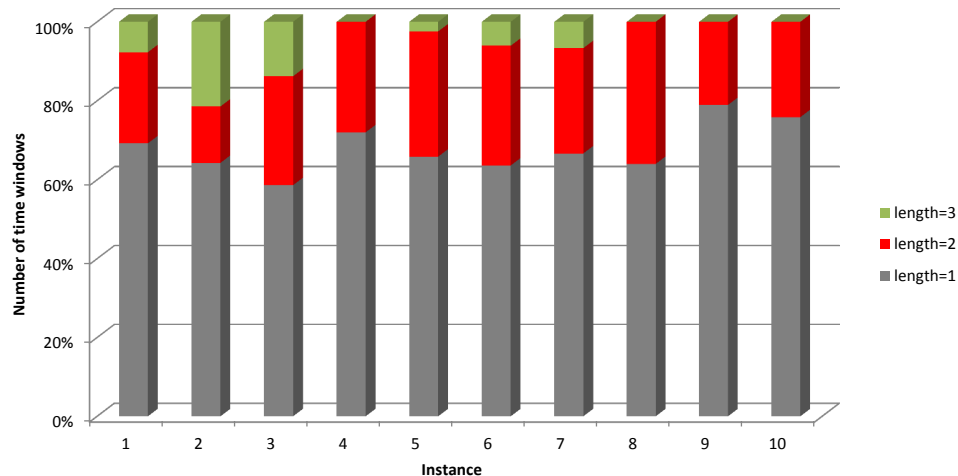


Figure 6: Phase II: Time window distribution

In Table 2, we report the solution times (in seconds) if solved to optimality, values of the best solutions found by the heuristic (if applied), primal values and dual bounds given by the full model, and optimality gaps. For each instance, we show the number of scenarios (including the scenario with no disruptions) in the second column. A limit of 4 hours is given to the full model. For each instance, there are three rows in the table, which correspond to the cases where (1) we solve the full model without the heuristic, (2) we use the heuristic with $K = 1$ before solving the full model, and (3) we use the heuristic with $K = 2$ before solving the full model, respectively. By using the two-phase stochastic multi-scenario construction heuristic (either with $K = 1$ or $K = 2$), we achieve a better performance of the full model, and for all the instances, the best solution obtained is at least as good as the best solution found by solving the full model without the construction heuristic. Another important observation is that as we increase the number of iterations at the improvement stage of the heuristic, we improve the quality of solutions in 8 out of the 10 instances; in one of the other two, an optimal solution to the full problem is readily obtained by using the heuristic with $K = 1$. However, in terms of the overall performance, it is unclear whether applying one or two improvement iterations of the construction heuristic is better.

Ins.	No. of scenarios	Best solution (CH)	Best solution (Full)	Bound	Gap	Time
1	15	–	30	30	–	315.7
		39	30	30	–	502.07
		31	30	30	–	78.8
2	15	–	92	92	–	77.3
		92	92	92	–	49.8
		92	92	92	–	20.7
3	30	–	132	122.4	7.25%	–
		133	132	132	–	10720.0
		132	132	132	–	5131.8
4	26	–	92	80.4	12.62%	–
		95	92	75.8	17.66%	–
		92	92	92	–	11596.0
5	42	–	419	119.8	71.40%	–
		171	171	134.9	21.11%	–
		171	171	123.7	27.65%	–
6	34	–	138	84.3	38.94%	–
		148	128	111.2	13.09%	–
		133	128	103.4	19.25%	–
7	31	–	103	52.1	49.44%	–
		93	82	67.5	17.72%	–
		91	91	61.30	32.65%	–
8	26	–	58	58	–	8049.6
		111	58	58	–	1592.9
		67	58	58	–	3951.0
9	20	–	31	31	–	1002.8
		33	31	31	–	292.6
		31	31	31	–	260.0
10	30	–	55	55	–	14180.3
		78	55	55	–	3520.7
		63	55	55	–	9591.6

Table 2: Phase II: Computational performances of the construction heuristic

6 Conclusions

In this paper, we study a robust maritime inventory routing problem with delivery time windows and stochastic travel times, where the length and placement of the time windows are also decision variables. We cast the problem as a two-stage stochastic mixed-integer program and propose a two-phase solution approach that considers a sample set of disruptions as well as their recovery solutions. Two planning strategies are proposed to generate robust routes, and a multi-scenario construction heuristic is introduced when we determine the placement of time windows.

The problem of determining the length and placement of time windows has largely been overlooked in the vehicle/inventory routing literature, and to the best of our knowledge, this paper is the first one to consider such a problem as it arises in inventory routing. We believe that time windows play an important role in helping a vendor design and negotiate long-term delivery contracts with its customers. By using an integrated solution procedure that simultaneously considers the routing and time window allocation, we generate committed delivery schedules and routing solutions that are robust against unplanned disruptions and have a low cost of achieving the robustness.

Acknowledgments

We wish to thank ExxonMobil Research and Engineering Company for financial support of this research.

References

- Agra, Agostinho, Marielle Christiansen, Rosa Figueiredo, Lars Magnus Hvattum, Michael Poss, Cristina Requejo. 2012. Layered formulation for the robust vehicle routing problem with time windows. A. Ridha Mahjoub, Vangelis Markakis, Ioannis Milis, Vangelis Th. Paschos, eds., *Combinatorial Optimization*. Springer, 249–260.
- Agra, Agostinho, Marielle Christiansen, Rosa Figueiredo, Lars Magnus Hvattum, Michael Poss, Cristina Requejo. 2013. The robust vehicle routing problem with time windows. *Computers & Operations Research* **40**(3) 856–866.
- Alvarez, J Fernando, Panagiotis Tsilingiris, Erna S Engebretsen, Nikolaos MP Kakalis. 2011. Robust fleet sizing and deployment for industrial and independent bulk ocean shipping companies. *INFOR: Information Systems and Operational Research* **49**(2) 93–107.
- Andersson, Henrik, Marielle Christiansen, Kjetil Fagerholt. 2010a. Transportation planning and inventory management in the lng supply chain. Endre Bjørndal, Mette Bjørndal, Panos M. Pardalos, Mikael Rönnqvist, eds., *Energy, natural resources and environmental economics*. Springer, 427–439.
- Andersson, Henrik, Arild Hoff, Marielle Christiansen, Geir Hasle, Arne Løkketangen. 2010b. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research* **37**(9) 1515–1536.
- Christiansen, Marielle. 1999. Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science* **33**(1) 3–16.
- Christiansen, Marielle, Kjetil Fagerholt. 2002. Robust ship scheduling with multiple time windows. *Naval Research Logistics (NRL)* **49**(6) 611–625.
- Christiansen, Marielle, Kjetil Fagerholt, Bjørn Nygreen, David Ronen. 2006. Maritime transportation. *Transportation* **14** 189–284.
- Christiansen, Marielle, Kjetil Fagerholt, Bjørn Nygreen, David Ronen. 2013. Ship routing and scheduling in the new millennium. *European Journal of Operational Research* **228**(3) 467–483.
- Christiansen, Marielle, Bjørn Nygreen. 1998. A method for solving ship routing problems with inventory constraints. *Annals of Operations Research* **81** 357–378.
- Coelho, Leandro C, Jean-François Cordeau, Gilbert Laporte. 2013. Thirty years of inventory routing. *Transportation Science* **48**(1) 1–19.
- Fagerholt, Kjetil. 2001. Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research* **131**(3) 559–571.
- Goel, Vikas, Kevin C Furman, Jin-Hwa Song, Amr S El-Bakry. 2012. Large neighborhood search for LNG inventory routing. *Journal of Heuristics* **18**(6) 821–848.
- Grønhaug, Roar, Marielle Christiansen. 2009. Supply chain optimization for the liquefied natural gas business. Jo A.E.E. Nunen, M. Grazia Speranza, Luca Bertazzi, eds., *Innovations in distribution logistics*. Springer, 195–218.

- Grønhaug, Roar, Marielle Christiansen, Guy Desaulniers, Jacques Desrosiers. 2010. A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science* **44**(3) 400–415.
- Halvorsen-Weare, Elin E, Kjetil Fagerholt, Mikael Rönnqvist. 2013. Vessel routing and scheduling under uncertainty in the liquefied natural gas business. *Computers & Industrial Engineering* **64**(1) 290–301.
- Jabali, Ola, Roel Leus, Tom Van Woensel, Ton de Kok. 2013. Self-imposed time windows in vehicle routing problems. *OR Spectrum* 1–22.
- Papageorgiou, Dimitri J, Ahmet B Keha, George L Nemhauser, Joel Sokol. 2014a. Two-stage decomposition algorithms for single product maritime inventory routing. *INFORMS Journal on Computing* **26**(4) 825–847.
- Papageorgiou, Dimitri J, George L Nemhauser, Joel Sokol, Myun-Seok Cheon, Ahmet B Keha. 2014b. Mirplib—a library of maritime inventory routing problem instances: Survey, core model, and benchmark results. *European Journal of Operational Research* **235**(2) 350–366.
- Rakke, Jørgen Glomvik, Henrik Andersson, Marielle Christiansen, Guy Desaulniers. 2014. A new formulation based on customer delivery patterns for a maritime inventory routing problem. *Transportation Science* .
- Rakke, Jørgen Glomvik, Magnus Stålhane, Christian Rørholt Moe, Marielle Christiansen, Henrik Andersson, Kjetil Fagerholt, Inge Norstad. 2011. A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. *Transportation Research Part C: Emerging Technologies* **19**(5) 896–911.
- Song, Jin-Hwa, Kevin C Furman. 2013. A maritime inventory routing problem: Practical approach. *Computers & Operations Research* **40**(3) 657–665.
- Spliet, Remy, Adriana F. Gabor. 2014. The time window assignment vehicle routing problem. *Transportation Science* doi:10.1287/trsc.2013.0510. URL <http://dx.doi.org/10.1287/trsc.2013.0510>.
- Stålhane, Magnus, Jørgen Glomvik Rakke, Christian Rørholt Moe, Henrik Andersson, Marielle Christiansen, Kjetil Fagerholt. 2012. A construction and improvement heuristic for a liquefied natural gas inventory routing problem. *Computers & Industrial Engineering* **62**(1) 245–255.
- Tirado, Gregorio, Lars Magnus Hvattum, Kjetil Fagerholt, Jean-François Cordeau. 2013. Heuristics for dynamic and stochastic routing in industrial shipping. *Computers & Operations Research* **40**(1) 253–263.
- Zhang, Chengliang, George Nemhauser, Joel Sokol, Myun-Seok Cheon, Ahmet Keha. 2013. Flexible solutions to maritime inventory routing problems with delivery time windows. Submitted for publication.