Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 176 (2007) 106–130
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

# Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model

Faiz Al-Khayyal [a,*,1], Seung-June Hwang [b,1]

[a] *School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, United States*
[b] *Department of Business Administration, Hanyang University, Ansan, Gyeonggi, 426-791, Korea*

Received 19 April 2004; accepted 22 June 2005
Available online 7 October 2005

**Abstract**

This paper formulates a model for finding a minimum cost routing in a network for a heterogeneous fleet of ships engaged in pickup and delivery of several liquid bulk products. The problem is frequently encountered by maritime chemical transport companies, including oil companies serving an archipelago of islands. The products are assumed to require dedicated compartments in the ship. The problem is to decide how much of each product should be carried by each ship from supply ports to demand ports, subject to the inventory level of each product in each port being maintained between certain levels that are set by the production rates, the consumption rates, and the storage capacities of the various products in each port. This important and challenging inventory constrained multi-ship pickup–delivery problem is formulated as a mixed-integer nonlinear program. We show that the model can be reformulated as an equivalent mixed-integer linear program with special structure. Over 100 test problems are randomly generated and solved using CPLEX 7.5. The results of our numerical experiments illuminate where problem structure can be exploited in order to solve larger instances of the model. Part II of the sequel will deal with new algorithms that take advantage of model properties.


*Keywords:* Maritime transportation; Liquid bulk; Multi-commodity; Petrochemical logistics; Inventory dependent time windows; Ship routing and scheduling

* Corresponding author. Tel.: +1 404 864 3037; fax: +1 404 894 2301.
*E-mail addresses:* faiz@isye.gatech.edu (F. Al-Khayyal), sejhwang@isye.gatech.edu (S.-J. Hwang).
[1] Supported by grants from The Logistics Institute Asia Pacific (TLI-AP) and the US National Science Foundation DMI-990826.

0377-2217/$ - see front matter 
doi:10.1016/j.ejor.2005.06.047

## 1. Introduction

This paper addresses the problem of determining a minimum cost routing schedule for a heterogeneous fleet of ships engaged in pickup and delivery of various liquid bulk products across a set of supply and demand harbors with specified product availabilities and needs, respectively. Due to the nature of the products, it is impossible to carry more than two products without being separated into dedicated compartments of the ships. The optimal routing schedule should specify how much of each product to carry from which port to which port, at what time, and on which ship, subject to the conditions that all ports must have sufficient product for consumption, and the stock levels of the products cannot exceed the inventory capacity of that port.

This problem is motivated by a real logistics problem faced by an oil company in Asia Pacific serving an archipelago of islands. This company has a fleet of tankers and barges that transport petrochemical products between various plants and has many storage terminals and direct customers. Since plants and customers are dispersed over many islands, and since there is no terrestrial transportation infrastructure, such as a pipeline network connecting the islands, it is necessary to carry all inter-island supply and demand by ships. Each island has a different production and consumption rate for specified products, and the inter-island transport schedule should be such that proper stock levels for the petrochemicals are maintained at each island during the planning horizon. The problem is further complicated by the fact that the ships are able to carry a number of different products at the same time, and since some of these products cannot mix, these need to be carried in separate dedicated compartments. Fig. 1 illustrates the problem for eight harbors, four products, and three ships in the Philippines.

In this paper, we first identify the most important logistics considerations for this difficult ship-routing problem. Next, using a network flow model, we formulate the problem as a combined multi-ship pickup–delivery problem. The interaction between multiple ships arriving at the same destination makes the formulation bilinearly constrained. We use novel linearization schemes to develop an *equivalent* mixed-integer linear programming reformulation for the problem.

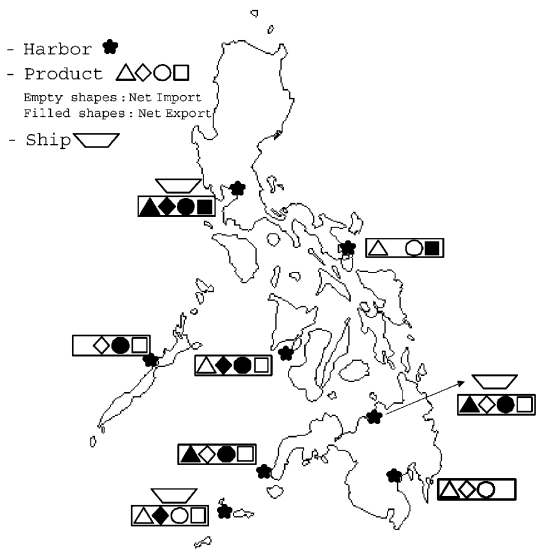


Fig. 1. A 4 product problem with 8 harbors and 3 ships.

The remainder of this paper is organized as follows. In Section 2, we review the existing literature on ship routing problems. Section 3 describes the critical characteristics of the problem under consideration. Section 4 develops an optimization model for the problem, and Section 5 presents equivalent linear reformulations of the nonlinear constraints in this model. Section 6 solves an illustrative example using commercial optimization software, and reports on numerical experiments which measure the sensitivity of solution time to changes in model parameters. Finally, Section 7 offers some concluding remarks. Two appendices include a notational summary and the proof of a key result.

## 2. Maritime routing and scheduling literature

Operations Research has long recognized the need for systematic mathematical techniques for the optimal routing and scheduling of vehicles to meet the needs of a dispersed set of customers. Models and solution algorithms for these so-called *vehicle routing* (cf., [20,11]) problems have revolutionized the operations of trucking industries. However, even though approximately 90% of the volume and 70% of the value of all goods transported worldwide is carried by sea [16], until recently, relatively little work has been done on optimization based routing and scheduling of ships. In this section, we review some of the existing work done in this area.

In general, maritime routing and scheduling problems for bulk products can be separated into *cargo routing* and *inventory routing* problems. Cargo routing problems are mainly constrained by the cargo, which is usually specified by the loading/discharging ports, and by time windows for loading and unloading. Inventory routing problems are constrained by inventory requirements such that the inventory level of products at ports should be maintained. Most cargo routing problems employ a set partitioning approach as a solution strategy. The approach first generates a set of candidate schedules and then chooses the best among these schedules. Below, we first survey the literature on cargo routing in order to illustrate the range of models and solution strategies that have been considered. This is followed by inventory routing models and solution methods, which are more closely related to the model studied herein.

Ronen [17] addresses a problem of scheduling the shipment of large quantities of bulk or semi-bulk products from one origin area to many destination ports. Vessels with different capacities deliver shipments to their destinations and return back to a common origin. The model does not consider any time window constraints because shipments are specified by their sizes and destinations. The paper proposes and compares three different algorithms: a single step cost minimization heuristic, a biased random search heuristic that chooses the cheapest schedule out of many generated schedules, and an optimization algorithm based on a mixed-binary nonlinear formulation.

Brown et al. [4] discuss the problem of routing and scheduling a fleet of tankers of similar sizes that carry crude oil from the Middle East to Europe and to North America. A voyage has a single loading port and a single discharging port and the cargo is a full shipload. The paper explicitly considers constraints on loading/discharging durations (time windows) for each cargo. The authors propose an enumerative method that generates all feasible schedules and select the best set of schedules. Later, Fisher and Rosenwein [12] address an extension to the problem in [4] whereby each cargo consists of a designated quantity of a product to be picked up from one or more load ports and delivered to one or more destination ports within specified time windows. The authors use a set partitioning approach similar to Brown et al. [4].

Baush et al. [3] discuss the distribution of multiple liquid bulk products among plants, distribution centers, and industrial customers by using vessels equipped with multiple compartments over a planning horizon of a few weeks. The model assumes all loads at known locations are pre-assigned to vessels and it specifies the earliest and latest loading dates for each product. The authors generate all feasible schedules for all vessels and choose the best schedule for each vessel.

Fagerholt and Christiansen [9] consider a combined multi-ship pick up and delivery problem with time windows and multi-compartments for *dry* bulk. Each ship in the fleet is equipped with a flexible cargo hold that can be partitioned into several small holds in a given number of ways. Consequently, multiple products can be delivered by the same ship at the same time. A set partitioning approach is proposed as a solution method. The technique incorporates a method detailed in Fagerholt and Christiansen [10] for solving a key subproblem.

The problems in [3,9] are related to the model in this paper in that they consider delivering multiple products that are carried in multiple compartments. These models require only a single pickup and delivery of products with given origin–destination pairs for each. The challenge is to accomplish this within specified pickup and delivery time windows for each product. In contrast, the model studied in this paper allows multiple pickups and deliveries of many products during a planning horizon, while maintaining specified inventory levels of products at all ports.

While the foregoing problems were solvable by a set partitioning approach, more efficient methods for related problems have also been proposed. Papadakis and Perakis [15] discuss the problem of a fleet of ships carrying a specific amount of a single bulk product to several destination ports during a specified time interval. Each vessel in the fleet loads at an origin, unloads at a destination and returns to its origin. The paper explicitly considers optimal speed selection for the ships. The solution method is based on decoupling the speed selection problem from the vessel allocation problem using Lagrangian relaxation. Jetlund and Karimi [13] focus on ship operation decisions under given shipment schedules. They seek the maximum profit scheduling for a fleet of ships delivering multiple liquid bulk products. Each product needs to be delivered from the pick-up port to the discharge port within time windows. A heuristic method to solve the problem yielded improved schedules over those being used by a multi-national shipping company in the Asia Pacific region.

The first maritime inventory routing problem was introduced by Miller [14]. The author describes a fleet scheduling and inventory resupply problem faced by an international chemical company that transfers multiple chemicals from one origin to multiple destinations under the condition that certain inventory levels are maintained. An interactive solution method was proposed, which utilizes both manual and automatic techniques to arrive at a good solution.

Christiansen and Nygreen [7] present a combined inventory management problem and ship routing problem with time windows. A fleet of ships transport a single product between production and consumption harbors. The quantities loaded and discharged are determined by the production rates of the harbors, possible stock levels, and the actual ships visiting the harbors. This problem is formulated as a mixed integer linear programming problem and solved by a method that combines Dantzig–Wolf decomposition with branch-and-bound. The columns needed by the Dantzig–Wolf method were generated using the scheme of Christiansen and Nygreen [8]. Later, Christiansen [5] reformulated this problem as a network model with side constraints and solved it using the method proposed by Christiansen and Nygreen [7].

Our ship routing problem is an extension to the model considered by Christiansen and Nygreen [7], and adopts the network flow formulation of Christiansen [5]. As in these papers, we consider inventory constrained scheduling of a heterogeneous fleet of ships, where there is no central source of supply. On the other hand, the major difference is that we consider pickup and delivery of *multiple products* using ships with *multiple dedicated compartments*. Additionally, we allow more than one ship to be docked in the same harbor at any given time, and impose inventory dependant pickup and delivery time windows.

Our brief review categorized the literature into cargo and inventory routing problems. A more thorough review by Christiansen et al. [6] classified models for tramp, liner, industrial, and military operations.

## 3. Multi-commodity bulk shipping

In this section we describe some of the critical characteristics of the multi-commodity bulk shipping problem under consideration.

We consider a heterogeneous fleet of ships and barges. The ships have multiple dedicated compartments that are able to carry different products simultaneously. The ships in the fleet differ by size, number of compartments, and the set of products they can carry.

The fleet is used to distribute multiple liquid bulk products amongst geographically dispersed ports. Each port is either a producer or a consumer for a certain product, and the average production and consumption rate for each product is known. A ship loads a product from a producing port or harbor, and unloads it at a consuming harbor. Partial loading of the ship is allowed.

The loading and unloading of a ship at a harbor is carried out in one of the piers or jetties. It is assumed that each harbor has enough piers to accommodate all incoming ships. There is no dedicated pier for any product type. However it is impossible to *simultaneously* load or unload different products onto a ship at a pier. Furthermore, more than two ships *cannot* be simultaneously loading and/or unloading the same product.

Under the above conditions, our problem is to determine (i) which product is to be loaded into (or unloaded from) which ship, (ii) the quantity to be loaded/unloaded, (iii) the time period of loading/unloading, and to schedule the arrivals and departures of the ships so as to maintain the inventory levels between operating bounds during the planning horizon. The overall plan should minimize the total daily cost of the ships: fuel costs, port and canal dues, and loading and unloading charges over a finite planning horizon. Our model considers a captive fleet of ships and does not consider chartering vessels.

At the beginning of the $v$th planning horizon $T_v$, it is assumed that the starting position and the product for each ship is known. Finally, it is also assumed that each ship starts from some harbor in the beginning of the planning horizon and finishes in some harbor at the end of the planning horizon. However, in practice, we would deal with the situation that ships are enroute at the start and end of a planning horizon by implementing a rolling horizon concept as follows. At the start of the first planning horizon $T_1$, we generate dummy harbors (with no supply or demand) at the position of the ships in the middle of the sea. The problem is solved based on the current information (e.g., location of ships, inventory levels of products, etc.). At a specified time *prior* to the end of $T_1$, we begin a new planning horizon $T_2$ with routing decisions *from* all enroute dummy harbors replacing the destinations determined by the previous planning horizon $T_1$.

## 4. Model formulation

In this section, we describe a mathematical model for the problem under consideration. The model is developed along the lines of Christiansen [5], but with significant modifications to account for multiple products, dedicated ship compartments and multi-ship port calls with overlapping docking times. In the following formulation, the decision variables are written in lower case letters and the parameters and sets are written in upper case letters. To keep the notation relatively simple, we assume that all ships are in ports at both the start and finish of the planning horizon; i.e., no ship is enroute at the beginning and end of the model's scheduling period.

### 4.1. Routing constraints

The routing constraints define and link the sequence of arrivals and departures of the various ships to and from various harbors. Let $V$ denote the set of all ships. Following Christiansen [5], let us define a network whose nodes are labeled $(i,m)$, where $i$ denotes a harbor, and $m$ is the arrival number at that harbor within the planning horizon. For example, node $(2,1)$ denotes the first arrival to harbor 2. We shall refer to such a pair as a *position*. Fig. 2 is an example network for 2 ships, and 3 harbors each having 2 positions. Let $H_T$ be the set of all harbors, and let $M_i$ be the set of possible arrival numbers $\{1,2,\ldots,\mu_i\}$ at harbor $i$

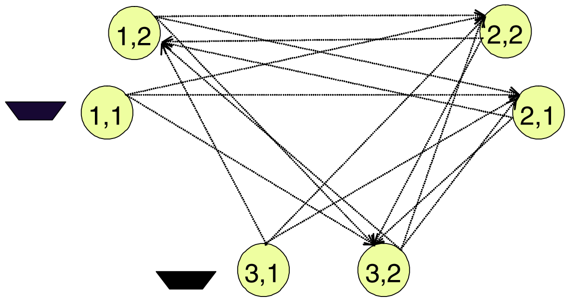


Fig. 2. Network model of 2 ships, 3 harbors with 2 positions for each harbor.

where $\mu_i$ is a specified number of arrivals to harbor $i$. We let $S_T$ denote the set of all feasible positions (harbor-arrival pairs); i.e., $S_T = \{(i,m) \mid i \in H_T,\, m \in M_i\}$. Let $(i_v, m_v) \in S_T$ denote the *initial position* of ship $v$. For example, if ship $v$ is initially located at harbor $i$, then $i_v = i$. If harbor $i$ initially has only one ship $v$, then $m_v = 1$. In cases when there are $\rho_i$ ships starting at harbor $i$, then arbitrarily set an arrival sequence number $m_v \in \{1, 2, \ldots, \rho_i\}$ for each ship $v \in \{v_1, v_2, \ldots, v_{\rho_i}\}$. Let $S_0 := \{(i_v, m_v) \mid v \in V\}$ be the set of initial positions for all ships. Then $S_N := S_T \backslash S_0$ is the set of all possible positions that ships can occupy *after* leaving their starting positions.

For $i \neq j$, for all $(i,m) \in S_N \cup \{(i_v, m_v)\}$, and $(j,n) \in S_N$, we set the binary variable $x_{imjnv}$ equal to 1 if ship $v \in V$ has a route segment that includes harbor $i$ as the $m$th arrival followed *immediately* by a visit to harbor $j$ as the $n$th arrival.

*4.1.1. Initial position constraints*

The following constraints enforce the requirement that each ship $v$ must depart from its initial position

$$\sum_{(j,n)\in S_N} x_{i_v m_v jnv} = 1, \quad \text{for every } v \in V. \tag{C1}$$

To allow for ships to remain unused in a harbor for the entire planning horizon, we introduce the binary variable $z_{imv}$ to equal 1 if ship $v$ ends its route as the $m$th arrival to harbor $i$. The constraint

$$\sum_{(j,n)\in S_N} x_{i_v m_v jnv} + z_{i_v m_v v} = 1, \quad \text{for every } v \in V,$$

ensures that ship $v$ will not depart its initial position whenever $z_{i_v m_v v} = 1$.

*4.1.2. Flow conservation constraints*

Flow conservation constraints ensure that the $m$th arrival to harbor $i$ should either leave harbor $i$ or end its route there. The flow conservation constraints

$$\sum_{(j,n)\in S_T} x_{jnimv} - \sum_{(j,n)\in S_N} x_{imjnv} - z_{imv} = 0, \quad \text{for every } (v,i,m) \in V \times S_N, \tag{C2}$$

guarantee that $z_{imv} = 0$ if $(i,m)$ is an intermediate position (non-initial position) and must equal to 1 if it is the final position of ship $v$'s schedule. This is because for each $(v,i,m)$, at most one ship $v$ can occupy position $(i,m)$ by the forthcoming constraints (C4); thus, $\sum_{(j,n)\in S_T} x_{jnimv} \leqslant 1$. If there is such an arrival, then it must depart unless harbor $i$ is the terminal point of the ship's journey during the planning horizon. In the latter case, $\sum_{(j,n)\in S_N} x_{imjnv} = 0$ so that constraint (C2) enforces $z_{imv} = 1$.

#### 4.1.3. Route finishing constraints

To simplify our notation, we will assume that at the beginning and end of the planning horizon, every ship is in a port and not enroute to some destination. Terminating journeys at a port can be achieved by imposing the constraints

$$\sum_{(i,m)\in S_N} z_{imv} = 1, \quad \text{for each } v \in V. \tag{C3}$$

When ships are allowed to stay at their initial positions, we need to replace $S_N$ by $S_N \cup \{(i_v, m_v)\}$ in constraint (C3).

#### 4.1.4. One time visit constraints

These constraints ensure each harbor-arrival pair $(i,m)$ is visited at most once. Let the binary variable $y_{im}$ equal to 1 when position $(i,m)$ is *not* visited. Then,

$$\sum_{v\in V}\sum_{(j,n)\in S_T} x_{jnimv} + y_{im} = 1, \quad \text{for every } (i,m) \in S_N, \tag{C4}$$

ensure that at most one ship can be the $m$th arrival to harbor $i$ and $y_{im}$ must be 1 when position $(i,m)$ is not visited.

#### 4.1.5. Arrival sequence constraints

Since it is not known a *priori* how many visits will be made to each harbor during a planning horizon, it is necessary to create enough positions $(i,m)$ to allow as many visits as needed for an optimal solution. Clearly, not all positions in every harbor will be utilized. However, if harbor $i$ does not have the $(m-1)$th arrival, then it cannot have the $m$th arrival; conversely, if there is an $m$th arrival, there must have been an $(m-1)$th arrival. This property can be expressed by the constraints

$$y_{im} - y_{i(m-1)} \geqslant 0, \quad \text{for every } (i,m) \in S_N. \tag{C5}$$

### 4.2. Constraints for loading and discharging

Constraints are needed to connect the quantities of various products to be loaded and unloaded at the various harbors to the capacities of the ships visiting these harbors. We introduce the following three sets of variables: $q_{imvk}$ corresponds to the quantity of product $k$ loaded onto or unloaded from ship $v$ at position $(i,m)$; $l_{imvk}$ corresponds to the quantity of product $k$ onboard ship $v$ as it departs from position $(i,m)$; and $o_{imvk}$ is a binary variable indicating whether product $k$ is loaded onto (or unloaded from) ship $v$ at position $(i,m)$. The following sets of parameters will also be used: $J_{ik}$ is equal to $+1$ (respectively, $-1$) if harbor $i$ is a producer (respectively, consumer) of product $k$, and 0 otherwise; $Q_{vk}$ is the quantity of product $k$ loaded onto ship $v$ at the start of the planning horizon; $CAP_{vk}$ is the capacity of the compartment onboard ship $v$ dedicated to carry product $k$. The set of all products that ship $v$ can carry is denoted by $K_v$.

#### 4.2.1. Ship load constraints

If a ship $v$ travels from position $(i,m)$ to position $(j,n)$, then the quantity $l_{jnvk}$ of product $k$ onboard at departure from $(j,n)$ should equal the quantity $l_{imvk}$ onboard at departure from $(i,m)$ plus, if $J_{jk} = +1$, (respectively, minus, if $J_{jk} = -1$) the quantity $q_{jnvk}$ loaded (respectively, unloaded) at $(j,n)$. But this will only happen if ship $v$ travels from $(i,m)$ to $(j,n)$; i.e., if $x_{imjnv} = 1$. Therefore, the loading constraints can be expressed as

$$x_{imjnv}[l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk}] = 0, \quad \text{for each } v \in V \text{ and every } (i,m,j,n,k) \in A_v \times K_v, \tag{1}$$

where $A_v := \{(i,m,j,n) \mid i \neq j, (i,m) \in S_N \cup S_0^v, (j,n) \in S_N\}$ is the set of all feasible arcs for ship $v$ in the network. The above constraints are nonlinear, but we will derive an *equivalent* linear system in Section 5.

#### 4.2.2. Initial ship load constraints

The amount $l_{i_v m_v vk}$ of product $k$ onboard ship $v$ at departure from the initial position $(i_v, m_v)$ should be equal to the initial quantity $Q_{vk}$ onboard plus if $J_{i_v k} = +1$ (respectively, minus if $J_{i_v k} = -1$) the quantity $q_{i_v m_v vk}$ loaded (respectively, unloaded) at the initial position. Thus,

$$Q_{vk} + J_{i_v k} q_{i_v m_v vk} - l_{i_v m_v vk} = 0, \quad \text{for each } v \in V \text{ and every } k \in K_v. \tag{C6}$$

#### 4.2.3. Compartment capacity constraints

The amount $l_{imvk}$ of product $k$ onboard ship $v$ at departure from position $(i,m)$ cannot exceed the capacity $CAP_{vk}$ of the compartment dedicated for product $k$. However, this will be meaningful only if ship $v$ visits $(i,m)$; i.e., $\sum_{(j,n)\in S_T} x_{jnimv} = 1$, otherwise the quantity $l_{imvk} = 0$. Therefore, the compartment capacity constraints can be expressed as

$$l_{imvk} \leqslant \sum_{(j,n)\in S_T} CAP_{vk} x_{jnimv}, \quad \text{for each } v \in V \text{ and every } (k,i,m) \in K_v \times S_N. \tag{C7}$$

#### 4.2.4. Servicing product constraints

Introduce the variable $o_{imvk}$ to indicate when product $k$ is serviced at position $(i,m)$ by ship $v$. We want $o_{imvk}$ to be 1 if $q_{imvk}$ is positive, otherwise it should be 0. That is, we want to ensure that the quantity $q_{imvk}$ of product $k$ loaded onto ship $v$ at position $(i,m)$ cannot exceed the capacity $CAP_{vk}$ of the compartment of ship $v$ dedicated for product $k$. This is expressed as

$$q_{imvk} \leqslant CAP_{vk} o_{imvk}, \quad \text{for each } v \in V \text{ and every } (k,i,m) \in K_v \times S_T. \tag{C8}$$

### 4.3. Constraints for time aspects

Constraints are needed to define the arrival and departure times of the $m$th arrival to harbor $i$. The variables used are: $t_{im}$ is the time of the $m$th arrival to harbor $i$; and $t_{Eim}$ is the departure time of the $m$th arrival to harbor $i$ which equals the service ending time at position $(i,m)$. The following parameters are also used: $TQ_{ik}$ is the time required to load (unload) one unit amount of product $k$ at harbor $i$; $W_i$ is the set-up time required to service a product at harbor $i$ (for notational simplicity, we assume fixed set-up times for a port no matter what products are being serviced); and $T_{ijv}$ is the time required by ship $v$ to sail from harbor $i$ to harbor $j$ plus the set-up time required at harbor $j$ immediately preceding loading and unloading service times.

#### 4.3.1. Service time sequence constraints

Clearly, the $m$th visit should occur after the $(m-1)$th visit. That is,

$$t_{im} - t_{i(m-1)} \geqslant 0, \quad \text{for every } (i,m) \in S_N. \tag{C9}$$

#### 4.3.2. Service finishing time constraints

The time of departure for the $m$th arrival to harbor $i$ (namely, $t_{Eim}$) equals the $m$th arrival time ($t_{im}$) plus the time required to service all products ($\sum_{v\in V}\sum_{k\in K_v} TQ_{ik} q_{imvk}$) plus the fixed setup time $W_i$ (incurred $\sum_{v\in V}\sum_{k\in K_v} o_{imvk}$ times) for switching from one product to another. This constraint is given by

$$t_{im} + \sum_{v\in V}\sum_{k\in K_v} TQ_{ik}q_{imvk} + W_i\sum_{v\in V}\sum_{k\in K_v} o_{imvk} - t_{Eim} = 0, \quad \text{for every } (i,m)\in S_T. \tag{C10}$$

*4.3.3. Route and schedule compatibility constraints*

If ship $v$ travels from position $(i,m)$ to $(j,n)$—that is, $x_{imjnv} = 1$—then the arrival time $t_{jn}$ at $(j,n)$ is the sum of the departure time $t_{Eim}$ from $(i,m)$ and the travel time $T_{ijv}$ from harbor $i$ to harbor $j$ by ship $v$. Thus,

$$x_{imjnv}[t_{Eim} + T_{ijv} - t_{jn}] \leqslant 0, \quad \text{for each } v\in V \text{ and every } (i,m,j,n)\in A_v. \tag{2}$$

Notice that these constraints are only valid when the positions $(i,m)$ and $(j,n)$ are directly connected by ship $v$; i.e., when $x_{imjnv} = 1$. Constraints (2) are nonlinear, but we shall present *equivalent* linear reformulations (in Section 5) that are derived from global optimization theory.

*4.4. Constraints for the inventories*

Inventory constraints connect the required stock levels at the harbors to the quantities loaded onto and unloaded from the visiting ships. The following variables are used: $s_{imk}$ is the stock level of product $k$ in harbor $i$ at the time of the $m$th arrival; $s_{Eimk}$ is the stock level of product $k$ in harbor $i$ when the $m$th ship departs; and $p_{im}$ is a binary variable which is equal to zero if the $m$th and $(m-1)$th arrivals to harbor $i$ overlap; i.e., the $m$th ship arrives before the $(m-1)$th ship departs harbor $i$. The set $K_i^H$ represents the set of products that harbor $i$ produces and consumes. The parameters used here are as follows: $J_{ik}$ is set equal to $+1$ (respectively, $-1$) if harbor $i$ is a producer (respectively, consumer) of product $k$; $R_{ik} > 0$ is the production (if $J_{ik} = +1$) or consumption (if $J_{ik} = -1$) *rate* of product $k$ in harbor $i$; $T$ is the length of the planning horizon; $S_{MNik}$ is the minimum allowable stock level of product $k$ at harbor $i$ (safety stock); and $S_{MXik}$ is the maximum allowable stock level of product $k$ at harbor $i$ (production/deliveries must stop when this level is reached).

*4.4.1. Initial inventory constraints*

We classify the harbors into two groups: those that have ships and those that do not at the start of the planning horizon. For those harbors that do not have ships (namely, $H_N := H_T\backslash\{j\,|\,(j,m)\in S_0\}$) the stock level $s_{i1k}$ of product $k$ in harbor $i$ at the time of the first ship arrival is the amount $IS_{ik}$ of product $k$ in harbor $i$ at the start of the planning horizon plus the amount produced when $J_{ik} = +1$ (or minus the amount consumed when $J_{ik} = -1$) until the arrival $t_{i1}$ of the first ship; i.e.,

$$s_{i1k} = IS_{ik} + J_{ik}R_{ik}t_{i1}, \quad \text{for every } (i,k)\in H_N\times K_i^H. \tag{C11}$$

For those harbors that do have ships at the start of the planning horizon, $t_{i1} = 0$ so that $s_{i1k} = IS_{ik}$.

*4.4.2. Inventory level constraints*

Constraints are needed to measure the inventory level of product $k$ at harbor $i$ when the $m$th arrival departs. (Recall that we allow the simultaneous servicing of multiple ships in the same harbor.) Suppose now that there are two ships in harbor $i$ at the same time. Although the notation $(i,m)$ determines which of the ships arrived first, it is not clear which ship leaves first. This can cause difficulties in modeling the inventory constraints. To tackle this issue, we make the simplifying assumption that the second ship entering harbor $i$ will load or unload its quantity of product $k$ with complete knowledge of how much of the same product the first ship will be loading or unloading. So, even when the first ship completes its service later than the second ship, the stock levels $s_{im}$ and $s_{Eim}$ will always be within their bounds (see constraints (C15) and (4)).

For product $k$ in harbor $i$, if ship $v$ is the $m$th arrival, then the stock level $s_{Eimk}$ equals the level $s_{imk}$ before ship $v$ arrives less the amount $q_{imvk}$ loaded if $J_{ik} = +1$ (or plus the amount $q_{imvk}$ unloaded if $J_{ik} = -1$) plus the amount produced (if $J_{ik} = +1$) while ship $v$ is being loaded (or minus the amount consumed (when $J_{ik} = -1$) while ship $v$ is unloading) at the rate $R_{ik}$ during the time period $t_{Eim} - t_{im}$. These inventory constraints can be expressed as follows:

$$s_{imk} - \sum_{v \in V} J_{ik} q_{imvk} + J_{ik} R_{ik}(t_{Eim} - t_{im}) - s_{Eimk} = 0, \quad \text{for every } (i,m,k) \in S_T \times K_i^H. \tag{C12}$$

### 4.4.3. Stock level constraints

Constraints are needed to ensure that the stock levels of a product are consistent between successive arrivals to a harbor. If only a *single* ship is allowed in a harbor at any time during the planning horizon, the constraints can be simply stated as

$$s_{Ei(m-1)k} + J_{ik} R_{ik}(t_{im} - t_{Ei(m-1)}) - s_{imk} = 0, \quad \text{for every } (i,m,k) \in S_N \times K_i^H.$$

Now suppose that there are two ships in harbor $i$, which arrived as the $(m-1)$th and $m$th ship. It could easily be the case that the $m$th ship starts servicing a product $k$, before the $(m-1)$th ship begins its servicing of the same product. However, in our model, we make the simplifying assumption that the $m$th ship will load or unload product $k$ only after the $(m-1)$th ship has completed its loading or unloading of the same product. The two ship constraint becomes

$$s_{Ei(m-1)k} + J_{ik} R_{ik}[t_{im} - t_{Ei(m-1)}]p_{im} = s_{imk}, \quad \text{for every } (i,m,k) \in S_N \times K_i^H. \tag{3}$$

Here, $p_{im}$ is 0 if there are two or more ships in harbor $i$ during the $m$th arrival. Thus, if there are two ships, constraint (3) sets $s_{Ei(m-1)k} = s_{imk}$ so that overlapping does not cause problems. The following constraints force $p_{im}$ to take on the right 0 or 1 value

$$t_{im} - t_{Ei(m-1)} \geqslant [p_{im} - 1]T, \quad \text{for every } (i,m) \in S_N, \tag{C13}$$

$$[t_{im} - t_{Ei(m-1)}] \leqslant T p_{im}, \quad \text{for every} (i,m) \in S_N. \tag{C14}$$

We only need constraints for two ships because, by assumption, the ships will have products serviced consecutively in the order they arrive. The above constraints enforce $p_{im}$ to be equal to 0 if $[t_{im} - t_{Ei(m-1)}] < 0$ (overlapping) and equal to 1 if $[t_{im} - t_{Ei(m-1)}] \geqslant 0$. Appealing to well-known results from global optimization theory, *equivalent* linear representations of nonlinear constraints (3) are presented in Section 5.

### 4.4.4. Stock level bounds

At any position $(i,m)$, the stock level of product $k$ should be within the prescribed levels at the beginning and ending of service. Thus,

$$S_{MNik} \leqslant s_{imk} \leqslant S_{MXik}, \quad \text{for every } (i,m,k) \in S_T \times K_i^H, \tag{C15}$$

$$S_{MNik} \leqslant s_{Eimk} + J_{ik} R_{ik}(T - t_{Eim})(y_{i(m+1)} - y_{im}) \leqslant S_{MXik}, \quad \text{for every } (i,m,k) \in S_T \times K_i^H. \tag{4}$$

Constraint (4) considers the stock level of product $k$ not only at the end of each service but also at the end of the planning horizon. It has the term $(y_{i(m+1)} - y_{im})$ which is 1 if $(i,m)$ is the last position for harbor $i$; otherwise, 0. Recall that $y_{im} = 1$ if position $(i,m)$ is not visited; otherwise, 0. Therefore, the term $R_{ik}(T - t_{Eim})(y_{i(m+1)} - y_{im})$ is only activated when $(i,m)$ is the last position for harbor $i$. Equivalent linear representations of nonlinear constraints (4) are also presented in Section 5. Notice that $y_{im} = 1$ implies $y_{i(m+1)} = 1$ because of the arrival sequence constraint (C5).

## 4.5. Objective function

The objective of our ship routing and scheduling model is to minimize total operating costs over the planning horizon. The key cost components are the traveling costs, which include fuel and ship operating costs, and the loading/unloading costs, which include port operations, duties, etc. The parameter $C_{ijv}$ denotes the total traveling cost for a ship $v$ from harbor $i$ to harbor $j$, and $C_{Wik}$ is the fixed cost of loading or unloading product $k$ at harbor $i$. The cost function of the problem can then be expressed as follows:

$$\sum_{v\in V}\sum_{(i,m,j,n)\in A_v} C_{ijv}x_{imjnv} + \sum_{(i,m)\in S_T}\sum_{v\in V}\sum_{k\in K_v} C_{Wik}o_{imvk}. \tag{O}$$

The optimization model for our problem is to find $(x, y, z, l, q, o, t, t_E, s, s_E, p)$ that minimize (O) subject to linear constraints (C1)–(C15) and nonlinear constraints (1)–(4) as well as variable bounds on $(l, q, t, t_E, s, s_E)$ and binary integrality restrictions on $(x, y, z, o, p)$. We will define equivalent linear representations for constraints (1)–(4) to yield a mixed integer linear programming formulation for our model.

## 5. Linear reformulation

In this section, we linearize the nonlinear terms and reformulate the problem into an *equivalent* mixed-integer linear program.

### 5.1. Linearizing ship load constraints

The feasible region defined by ship load constraints (1) has the following general nonlinear structure

$$\{(x,y) \mid xf(y) = 0, x \in \{0,1\}, y \in \mathbb{Y}\}, \tag{5}$$

where $f(\cdot)$ is a function with domain $\mathbb{Y}$. Specifically, setting $x := x_{imjnv}$, $y := (l_{imvk}, l_{jnvk}, q_{jnvk})$, and $f(y) := l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk}$ in (5) yields constraint (1).

The constraint set given by (5) has a simpler characterization. First we need the following result.

**Proposition 1.** *Consider the set* $S := \{(x,y) \mid xf(y) = 0, x \in \{0,1\}, y \in \mathbb{Y}\}$, *where* $\{f(y) \mid y \in \mathbb{Y}\}$ *is compact; i.e., there exist bounds* $[L, U]$ *such that* $L \leqslant f(y) \leqslant U$ *for all* $y \in \mathbb{Y}$. *Then, set S is equivalent to*

$$S' := \{(x,y) \mid L(1-x) \leqslant f(y) \leqslant U(1-x), x \in \{0,1\}, y \in \mathbb{Y}\}.$$

**Proof.** The proof of the above result is straightforward and omitted. □

For constraint (1), $f(y) := l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk}$ is linear and $-CAP_{vk}$ and $CAP_{vk}$ are valid lower and upper bounds. Using Proposition 1, we can then replace (1) with the equivalent linear constraints

$$l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk} + CAP_{vk}x_{imjnv} \leqslant CAP_{vk}, \quad \text{for every } v \in V, \text{ and every } (i,m,j,n,k) \in A_v \times K_v, \tag{C16}$$

$$l_{imvk} + J_{jk}q_{jnvk} - l_{jnvk} - CAP_{vk}x_{imjnv} \geqslant -CAP_{vk}, \quad \text{for every } v \in V, \text{ and every } (i,m,j,n,k) \in A_v \times K_v. \tag{C17}$$

### 5.2. Linearizing route and schedule compatibility constraints

Note that the route and schedule compatibility constraints (2) also have the same structure as (5). Here, setting $x := x_{imjnv}$, $y := (t_{Eim}, t_{jn})$, and $f(y) := t_{Eim} + T_{ijv} - t_{jn}$ in (5) gives constraint (2). In this case, the upper bound on $f(y)$ is $2T$. Notice that (2) are inequality constraints. Using Proposition 1, we can replace (2) with the equivalent linear constraint

$$t_{Eim} + T_{ijv} - t_{jn} + 2Tx_{imjnv} \leqslant 2T, \quad \text{for every } v \in V, \text{ and every } (i,m,j,n) \in A_v. \tag{C18}$$

### 5.3. Linearizing stock level constraints

The stock level constraints (3) given by

$$s_{Ei(m-1)k} + J_{ik}R_{ik}[t_{im} - t_{Ei(m-1)}]p_{im} = s_{imk}, \quad \text{for every } (i,m,k) \in S_N \times K_i^H$$

are linearized using the convex and concave envelopes of bilinear forms (see [2,1,19,18]).

The linearization process is accomplished in the following way. First, derive bounds for $(t_{im} - t_{Ei(m-1)})$. Noting that either a service time or the time between a departure and the next arrival can be as large as the entire planning horizon with the other quantity being small, we conclude that $-T \leqslant (t_{im} - t_{Ei(m-1)}) \leqslant T$. Next introduce a new variable $w_{im}$ in place of $[t_{im} - t_{Ei(m-1)}]p_{im}$, and replace (3) by the linear system of equations and inequalities (cf., [2,1])

$$s_{Ei(m-1)k} + J_{ik}R_{ik}w_{im} = s_{imk}, \quad \text{for every } (i,m,k) \in S_N \times K_i^H, \tag{6}$$
$$w_{im} \geqslant -Tp_{im}, \quad \text{for every } (i,m) \in S_N, \tag{7}$$
$$w_{im} \geqslant t_{im} - t_{Ei(m-1)} + Tp_{im} - T, \quad \text{for every } (i,m) \in S_N, \tag{8}$$
$$w_{im} \leqslant t_{im} - t_{Ei(m-1)} - Tp_{im} + T, \quad \text{for every } (i,m) \in S_N, \tag{9}$$
$$w_{im} \leqslant Tp_{im}, \quad \text{for every } (i,m) \in S_N, \tag{10}$$
$$p_{im} \in \{0,1\}, \quad \text{for every } (i,m) \in S_N. \tag{11}$$

**Remark.** The projection of the set defined by (6)–(11) onto the vector space determined by constraints (3) is a polyhedral outer approximation of the constraint region (3). This result follows from Proposition 3 by taking $x := p_{im}$ and $f(y) := t_{im} - t_{Ei(m-1)}$ together with $[L,U] = [-T,T]$, $\{l,u\} = \{0,1\}$ and $[a,b] = [-\infty,\infty]$. While (6)–(11) represent a polyhedral relaxation of (3), we show in Theorem 1 that, under optimization, our reformulation is exact; i.e., the optimal solution with linear constraints (6)–(11) is also optimal for the nonlinear model having constraints (3).

Alternatively, instead of linearizing $[t_{im} - t_{Ei(m-1)}]p_{im}$ with one variable $w_{im}$, we can consider linearizing the two terms $t_{im}p_{im}$ and $t_{Ei(m-1)}p_{im}$ separately by introducing two sets of variables, $w^1_{im}$ and $w^2_{im}$, respectively, in the following way. Both $t_{im}$ and $t_{Eim}$ are bounded below by 0 and above by $T$. Using these bounds, analogous to (6)–(11), we can replace (3) by the system of linear equations and inequalities

$$s_{Ei(m-1)k} + J_{ik}R_{ik}[w^1_{im} - w^2_{im}] = s_{imk}, \quad \text{for every } (i,m,k) \in S_N \times K_i^H \tag{C19.a}$$
$$w^1_{im} \geqslant 0, \quad \text{for every } (i,m) \in S_N, \tag{C19.b}$$
$$w^1_{im} \geqslant t_{im} + Tp_{im} - T, \quad \text{for every } (i,m) \in S_N, \tag{C19.c}$$
$$w^1_{im} \leqslant t_{im}, \quad \text{for every } (i,m) \in S_N, \tag{C19.d}$$
$$w^1_{im} \leqslant Tp_{im}, \quad \text{for every } (i,m) \in S_N, \tag{C19.e}$$
$$w^2_{im} \geqslant 0, \quad \text{for every } (i,m) \in S_N, \tag{C19.f}$$
$$w^2_{im} \geqslant t_{Ei(m-1)} + Tp_{im} - T, \quad \text{for every } (i,m) \in S_N, \tag{C19.g}$$
$$w^2_{im} \leqslant t_{Ei(m-1)}, \quad \text{for every } (i,m) \in S_N, \tag{C19.h}$$
$$w^2_{im} \leqslant Tp_{im}, \quad \text{for every } (i,m) \in S_N. \tag{C19.i}$$
$$p_{im} \in \{0,1\}, \quad \text{for every } (i,m) \in S_N. \tag{C19.j}$$

Analogous to the first alternative, applying Proposition 3 twice to (3) yields the linear relaxations (C19.a)–(C19.j) which are exact under optimization by Theorem 1.

By virtue of Proposition 2, the reformulation obtained by linearizing terms $t_{im}p_{im}$ and $t_{Ei(m-1)}p_{im}$ separately using two variables is tighter than that obtained by linearizing $[t_{im} - t_{Ei(m-1)}]p_{im}$ using a single variable. While the two reformulations are equivalent when the integrality restriction on $p_{im}$ is included (cf., Theorem 1), the tighter reformulation is preferable from a computational viewpoint when the integrality restriction is relaxed. Note that the two reformulations define feasible sets in higher dimensions than the region defined by (3). When comparing tightness of relaxations we will always be looking at the projection of each relaxation onto the space of original variables; i.e., the space defined by (3).

Denote the continuous relaxation of (C19.j) as $\overline{\text{(C19.j)}}$ and (11) as $\overline{(11)}$; i.e., both $\overline{\text{(C19.j)}}$ and $\overline{(11)}$ label the conditions $0 \leqslant p_{im} \leqslant 1$ for every $(i,m) \in S_N$.

**Proposition 2.** *For each point feasible to (C19.a)–(C19i) and $\overline{\text{(C19.j)}}$, there is a corresponding point feasible to the continuous relaxation of (6)–(10) and $\overline{(11)}$.*

**Proof.** This result is stated and proved in a more general setting in Appendix B. $\square$

**Remark.** Let $S_1$ be the projection of the set defined by (C19.a)–(C19.i) and $\overline{\text{(C19.j)}}$ onto the vector space of the feasible set of the general model in Section 4 given by (C1)–(C15) and (1)–(4). Now let $S_2$ be the projection of the continuous relaxation of (6)–(10) and $\overline{(11)}$ onto the same vector space. Then, by Proposition 2, we have $S_1 \subseteq S_2$; i.e., (C19) yields a tighter relaxation of (3).

We next show why our linear reformulations are exact under optimization. First, we need the following results.

**Proposition 3.** *Consider the nonlinear feasible region $P_1$, where $L \leqslant U$ and $l \leqslant u$, and the relaxation $P_2$ defined as*

$$\begin{aligned} P_1 &:= \{(x,y) \mid a \leqslant xf(y) \leqslant b, L \leqslant f(y) \leqslant U, x \in \{l,u\}\} \\ P_2 &:= \{(x,y,z) \mid a \leqslant z \leqslant b, L \leqslant f(y) \leqslant U, x \in \{l,u\}, z \geqslant lf(y) + Lx - Ll, z \geqslant uf(y) + Ux - Uu, \\ &\qquad z \leqslant uf(y) + Lx - Lu, z \leqslant lf(y) + Ux - Ul\}. \end{aligned}$$

*If $(x,y,z) \in P_2$, then $z = xf(y)$ and $(x,y) \in P_1$.*

**Proof.** There are two cases

*Case 1.* $(x = l)$ We have from $P_2$

$$\begin{aligned} z &\geqslant lf(y) + Ll - Ll \;\Rightarrow\; z \geqslant lf(y), \\ z &\geqslant uf(y) + Ul - Uu \;\Rightarrow\; z - uf(y) \geqslant U(l-u), \\ z &\leqslant uf(y) + Ll - Lu \;\Rightarrow\; z - uf(y) \leqslant L(l-u), \\ z &\leqslant lf(y) + Ul - Ul \;\Rightarrow\; z \leqslant lf(y). \end{aligned}$$

Thus, $z = lf(y)$ and $U(l-u) \leqslant z - uf(y) \leqslant L(l-u) \;\Rightarrow\; L \leqslant f(y) \leqslant U$ because $l \leqslant u$ and $z - uf(y) = f(y)(l-u)$.

*Case 2.* $(x = u)$ We have from $P_2$

$$\begin{aligned} z &\geqslant lf(y) + Lu - Ll \;\Rightarrow\; z - lf(y) \geqslant L(u-l), \\ z &\geqslant uf(y) + Uu - Uu \;\Rightarrow\; z \geqslant uf(y), \\ z &\leqslant uf(y) + Lu - Lu \;\Rightarrow\; z \leqslant uf(y), \\ z &\leqslant lf(y) + Uu - Ul \;\Rightarrow\; z - lf(y) \leqslant U(u-l). \end{aligned}$$

Thus, $z = uf(y)$ and $L \leqslant f(y) \leqslant U$. Therefore, if $(x,y,z) \in P_2$, then $z = xf(y)$ and $(x,y) \in P_1$. $\square$

If $f(y)$ is discrete and $x$ is continuous, we have the analogous statement.

**Proposition 4.** *For given $L \leqslant U$ and $l \leqslant u$, define the sets*

$$P_1' := \{(x,y) \mid a \leqslant xf(y) \leqslant b,\ l \leqslant x \leqslant u,\ f(y) \in \{f(L), f(U)\}\}$$

$$\begin{aligned} P_2' := \{(x,y,z) \mid\ & a \leqslant z \leqslant b,\ l \leqslant x \leqslant u,\ f(y) \in \{f(L), f(U)\}, \\ & z \geqslant f(L)x + lf(y) - lf(L),\ z \geqslant f(U)x + uf(y) - uf(U), \\ & z \leqslant f(U)x + lf(y) - lf(U),\ z \leqslant f(L)x + uf(y) - uf(L)\}. \end{aligned}$$

*If $(x,y,z) \in P_2'$, then $z = xf(y)$ and $(x,y) \in P_1'$.*

We can now state the main result.

**Theorem 1.** *Let $(P)$ denote an optimization problem which has terms $xf(y)$, where $(x,y) \in P_1 \cup P_1'$, and let $(P_R)$ denote the corresponding relaxed problem obtained by replacing $xf(y)$ with $z$, $P_1$ with $P_2$, and $P_1'$ with $P_2'$. Then the $(x,y)$ component of the optimal solution of problem $(P_R)$ is optimal for problem $(P)$.*

**Proof.** Follows from Propositions 3 and 4. □

**Remark.** The relaxation of $(P)$ given by $(P_R)$ is exact in the sense that it will always produce an optimal solution for $(P)$. More precisely, if $(x^*, y^*, z^*)$ solves $(P_R)$, then $(x^*, y^*)$ solves $(P)$.

### *5.4. Linearizing stock level bounds constraints*

The stock level constraints (4) can be rewritten as

$$S_{MNik} \leqslant s_{Eimk} + J_{ik}R_{ik}T(y_{i(m+1)} - y_{im}) - J_{ik}R_{ik}t_{Eim}y_{i(m+1)} + J_{ik}R_{ik}t_{Eim}y_{im} \leqslant S_{MXik},$$
$$\text{for every } (i,m,k) \in S_T \times K_i^H.$$

We shall linearize this constraint using the reformulation technique explained in Section 5.3. Using Proposition 1, we linearize the terms $t_{Eim}y_{i(m+1)}$ and $t_{Eim}y_{im}$ by introducing the two sets of variables $u_{im}^1$ and $u_{im}^2$, respectively, together with the bounds $0 \leqslant t_{im}, t_{Eim} \leqslant T$. It follows that constraints (4) are equivalent (under optimization) to the system of linear inequalities

$$S_{MNik} \leqslant s_{Eimk} + J_{ik}R_{ik}(T)(y_{i(m+1)} - y_{im}) - J_{ik}R_{ik}u_{im}^1 + J_{ik}R_{ik}u_{im}^2 \leqslant S_{MXik},$$
$$\text{for every } (i,m,k) \in S_T \times K_i^H, \tag{C20.a}$$

$$u_{im}^1 \geqslant 0, \quad \text{for every } (i,m) \in S_T, \tag{C20.b}$$

$$u_{im}^1 \geqslant t_{Eim} + y_{i(m+1)} - T, \quad \text{for every } (i,m) \in S_T, \tag{C20.c}$$

$$u_{im}^1 \leqslant t_{Eim}, \quad \text{for every } (i,m) \in S_T, \tag{C20.d}$$

$$u_{im}^1 \leqslant y_{i(m+1)}, \quad \text{for every } (i,m) \in S_T, \tag{C20.e}$$

$$u_{im}^2 \geqslant 0, \quad \text{for every } (i,m) \in S_T, \tag{C20.f}$$

$$u_{im}^2 \geqslant t_{Eim} + y_{im} - T, \quad \text{for every } (i,m) \in S_T, \tag{C20.g}$$

$$u_{im}^2 \leqslant t_{Eim}, \quad \text{for every } (i,m) \in S_T, \tag{C20.h}$$

$$u_{im}^2 \leqslant y_{im}, \quad \text{for every } (i,m) \in S_T, \tag{C20.i}$$

$$y_{im} \in \{0,1\}, \quad \text{for every } (i,m) \in S_T. \tag{C20.j}$$

## 6. Mixed integer linear programming formulation

Combining all linear reformulations of the nonlinear constraints (1)–(4) with the linear constraints (C1)–(C15) yields the mixed integer linear program

$$\min_{(x,y,z,l,q,o,t,t_E,s,s_E,p,w^1,w^2,u^1,u^2)} \text{Objective function (O)}$$

subject to constraints (C1)–(C20.j)

$$t_{im} \leqslant T, \quad \text{for every } (i,m) \in S_T,$$

$$t_{Eim} \leqslant T, \quad \text{for every } (i,m) \in S_T,$$

$l, q, s, s_E, w^1, w^2, u^1, u^2$ nonnegative vectors (possibly with given upper bounds),

$x, y, z, o, p$ binary vectors.

Below, we solve a small illustrative example and vary some model parameters to gain some insights into developing solution algorithms that exploit problem structure.

### 6.1. Example

Consider the case of 2 ships ($V = \{1,2\}$) carrying 2 products ($K_v = \{1,2\}$, $\forall v \in V$) between 3 harbors ($H_T = \{1,2,3\}$), with each harbor handling both products (i.e., $K_i^H = \{1,2\}$ for $i = 1,2,3$). Furthermore, harbor 1 consumes product 1 ($J_{11} = -1$) and produces product 2 ($J_{12} = +1$), while harbors 2 and 3 both consume product 2 ($J_{i2} = -1$, for $i = 2,3$) and produce product 1 ($J_{i1} = +1$ for $i = 2,3$). We want to find the optimal ship routing for a 2 day planning horizon ($T = 2$). Assume that ship 1 is initially located in harbor 1 ($(i_1, m_1) = (1,1)$) with capacities $CAP_{11} = 10$ and $CAP_{12} = 10$ for products 1 and 2, respectively. Further assume that ship 2 is initially located in harbor 3 ($(i_2, m_2) = (3,1)$) with capacities $CAP_{21} = 10$ and $CAP_{22} = 25$ for products 1 and 2, respectively. Finally, assume that the compartments of both ships are initially empty ($Q_{vk} = 0$, $\forall v \in V$, $k \in K_v$). Initial inventory levels of all products ($IS_{ik}$, $\forall i \in H_T$, $k \in K_i^H$) at each harbor are given in Table 1, while the production rates ($R_{ik}$, $\forall i \in H_T$, $k \in K_i^H$) of all products at each harbor are listed in Table 2, where positive rates are production and negative rates are consumption. Assume that it takes 0.3 days to travel from one harbor to each of the others for each ship ($T_{ijv} = 0.3$, $\forall i,j \in H_T$, $i \neq j$, $v \in V$); however, the cost for traveling between the harbors is different for each ship. It costs \$1 for ship 1 to travel between any two harbors ($C_{ij1} = 1$, $\forall i,j \in H_T$, $i \neq j$) and it costs \$1.5 per trip to operate ship 2 ($C_{ij2} = 1.5$, $\forall i,j \in H_T$, $i \neq j$). The unit cost of loading or unloading any product at any harbor is taken as \$0.5 ($C_{Wik} = 0.5$, $\forall i \in H_T$, $k \in K_i^H$). The time it takes to service one unit of any product is assumed to be 0.01 days ($TQ_{ik} = 0.01$, $\forall i \in H_T$, $k \in K_i^H$), and set up times are taken as 0 ($W_i = 0$,

Table 1
Initial inventory levels $IS_{ik}$ for product $k$ in harbor $i$

| $IS_{11}$ | $IS_{12}$ | $IS_{21}$ | $IS_{22}$ | $IS_{31}$ | $IS_{32}$ |
|---|---|---|---|---|---|
| 10 | 15 | 5 | 15 | 10 | 15 |

Table 2
Daily rates $R_{ik}$ for product $k$ in harbor $i$

| $J_{11}R_{11}$ | $J_{12}R_{12}$ | $J_{21}R_{21}$ | $J_{22}R_{22}$ | $J_{31}R_{31}$ | $J_{32}R_{32}$ |
|---|---|---|---|---|---|
| −10 | 20 | 5 | −10 | 5 | −15 |

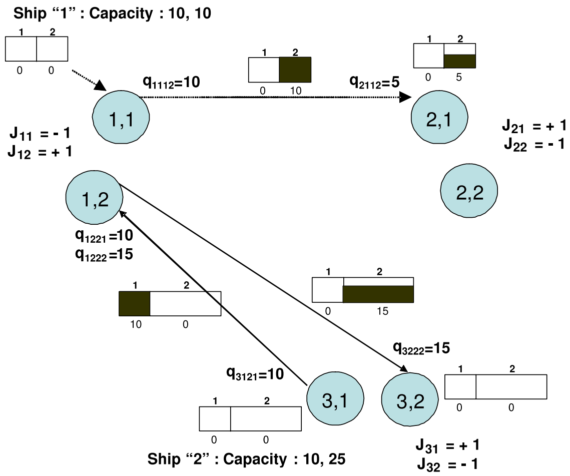


Fig. 3. A feasible route.

$\forall i \in H_T$). Fig. 3 shows a feasible route for this example, where each node represents a position (defined as ordered pairs of harbor and arrival numbers).

The feasible schedule illustrated in Fig. 3 determines the inventory levels over time of all products at each harbor, and this is displayed in Fig. 4 for our planning horizon of 2 days. In each chart, the solid line represents the change of inventory level as ships load or unload, while the dashed line represents the change of inventory if no loading or unloading occurs. For example, the chart for product 1 in harbor 1 shows that the initial inventory level starts from 10, and is consumed at the rate of 10. If no ship arrives (represented by a dashed line) before time 1.0, the stock is depleted by the end of the first day. However, before time 0.5, our feasible solution has the first ship unloading 10 units of product 1 so that the stock level is maintained between its upper (capacity limit) and lower levels, 20 and 0, respectively, during the planning horizon.

Notice that the inventory level of product 1 in harbor 2 increases because our feasible solution does not call for loading this product by any ship during the planning horizon. As can be seen, inventory levels of all products in all ports are maintained between their upper and lower levels.

The mixed-integer linear program for this problem has 384 constraints and 155 variables including 61 binary variables. It is solved optimally by ILOG CPLEX 7.500 in 0.02 seconds on a four-CPU Sun E450 server machine using only the default options of the solver. The optimal solution turns out to be the feasible solution displayed in Fig. 3. The total cost is \$7, consisting of \$4 for travel costs for the single trip of ship 1 and the two trips of ship 2, and \$3 for loading and unloading (ship 1 loads and unloads product 1, and ship 2 loads and unloads both products 1 and 2 for a total of six service calls costing \$0.5 each).

For the purpose of this illustrative example, we are not interested in devising the best CPLEX solution strategy. Rather, we seek to use the exact CPLEX solutions to uncover the problem parameters most sensitive to scaling, and to use this knowledge in developing a solution strategy for solving larger instances of the model. This will be presented in Part II of the sequel.

### 6.2. Computing time

Since it is not known *a priori* how many visits will be made to each harbor during a planning horizon, it is necessary to create enough positions $(i, m)$ to allow as many visits as needed for an optimal solution. However, the number of binary variables and the number of constraints in the model grow exponentially

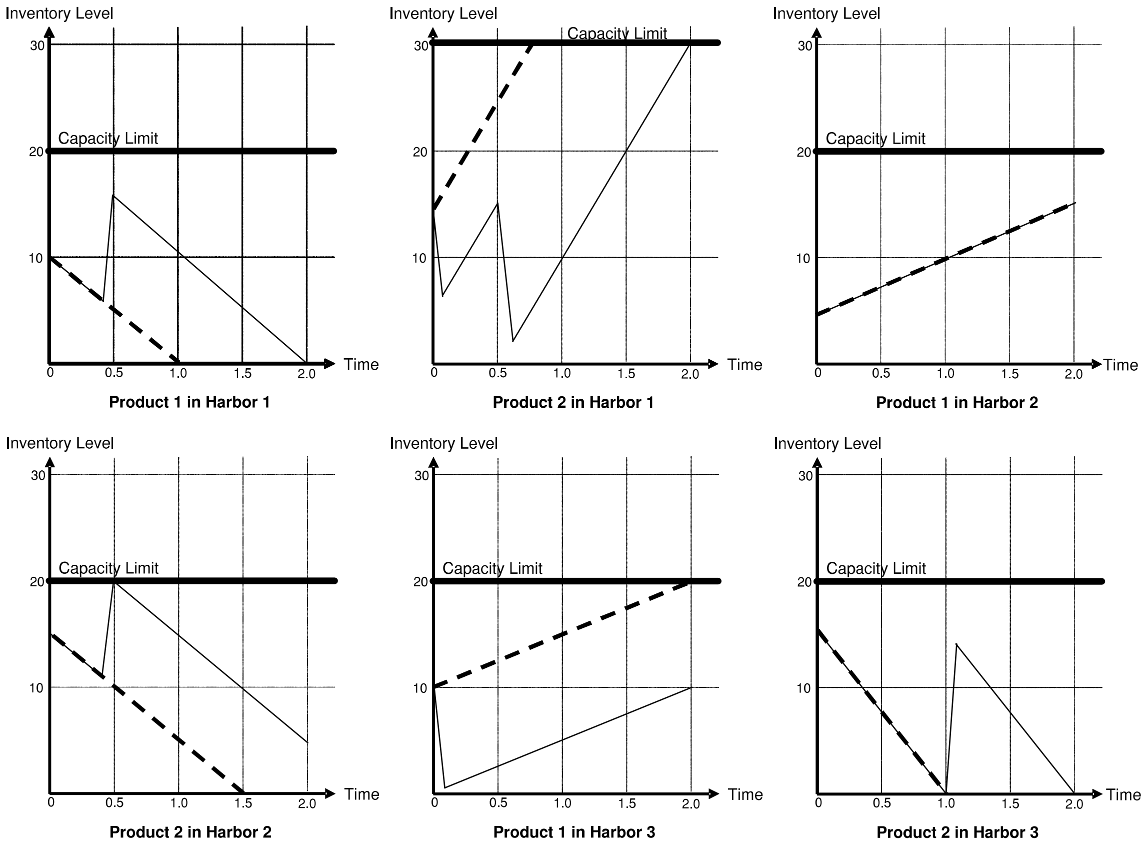


Fig. 4. Examples of movement of inventory levels of products at harbors.

with the number of positions (harbor-arrival pairs), while an insufficient number of positions may lead to an infeasible problem; i.e., a model with no feasible solutions. In this section, we first investigate the impact of the number of positions on computing time, and later show the relationship between the number of positions and the solution quality.

To measure the increase in solution time as a function of the number of positions in the model, a preliminary computational experiment was conducted on one-hundred randomly generated test problems. For our experiment, we chose ten different settings of the triple $(|V|, |H_T|, |K|)$, as listed in Table 3, and generated ten test problems for each setting.

For our test problems the $|H_T|$ harbors are first randomly located on the plane within a box. For each $i \in H_T$, the location of port $i$, denoted by $(a_i, b_i)$, is randomly generated by taking $a_i, b_i \sim U[0, 10]$, where $U[\alpha, \beta]$ denotes the uniform distribution over the interval $[\alpha, \beta]$. For simplicity, the distance between harbors $i$ and $j$ is assumed to be the Euclidian metric

Table 3
Ten test configurations (number of ships $|V|$, harbors $|H_T|$, and products $|K|$)

| | | | | |
|---|---|---|---|---|
| (3, 2, 2) | (3, 2, 3) | (3, 3, 2) | (3, 3, 3) | (4, 2, 2) |
| (4, 2, 3) | (4, 3, 2) | (4, 3, 3) | (4, 4, 2) | (4, 4, 3) |

Table 4
Generation of parameters for test problems

| Parameter | | Distribution/value |
|---|---|---|
| $T$ | Planning horizon | 10 |
| $C_{ijv}$ | Cost of travel form port $i$ to $j$ by ship $v$ | $w_v \times \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}$ |
| $T_{ijv}$ | Travel time between ports $i$ and $j$ by ship $v$ | $T/5 + 0.4\sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}/w_v$ |
| $CAP_{vk}$ | Capacity of product $k$ on ship $v$ | $U[20, 70]$ |
| $Q_{vk}$ | Initial quantity of product $k$ on ship $v$ | $CAP_{vk} \times U[0, 1]$ |
| $C_{Wik}$ | Fixed cost to service product $k$ in port $i$ | $U[5, 10]$ |
| $J_{ik}$ | $+1$, if port $i$ produces product $k$, $-1$, otherwise | either $+1$ or $-1$ with probability 1/2 |
| $R_{ik}$ | Rate of production or consumption of product $k$ in port $i$ | $U[1, 6]$ |
| $S_{MXik}$ | Maximum stock level of product $k$ in port $i$ | $U[20, 70]$ |
| $S_{MNik}$ | Minimum stock level of product $k$ in port $i$ | 0 |
| $IS_{ik}$ | Initial stock level of product $k$ in port $i$ | $S_{MXik} \times U[0.3, 0.7]$ |
| $TQ_{ik}$ | Time to load/unload product $k$ in port $i$ | $U[0, 0.03]$ |
| $W_i$ | Set-up time to change-over products in port $i$ | $U[0, 0.1]$ |

$$\|(a_i, b_i) - (a_j, b_j)\| = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}.$$

To differentiate between our vessels, we generate a weighting factor $w_v \sim U[0.5, 1]$ for each $v$, that influences the travel cost and travel time. In particular, travel cost ($C_{ijv}$) for vessel $v$ is taken to be proportional to travel distance with constant of proportionality $w_v$. On the other hand, as travel costs go up, we would expect travel times ($T_{ijv}$) to go down. This is accomplished by taking the simplifying assumption that travel time is proportional to travel distance with constant of proportionality $1/w_v$. The values for each problem's parameters were generated in accordance with Table 4. The ranges for the uniform distributions and scaling factors for travel times were selected to create nontrivial problems.

To complete the specification of our test problems, we need to fix the number of possible visits $\mu_i$ for harbor $i$. We need to choose $\mu_i$ large enough to admit an optimal solution, but not too large as to require long solution times.

To that end, we first determine the *minimum* number of visits to each harbor within the planning horizon. This minimum, $m_i$, for harbor $i$ can be calculated by considering the length of the planning horizon ($T$), the maximum and the minimum stock levels (respectively, $S_{MXik}$ and $S_{MNik}$) of product $k$ in harbor $i$, the capacity ($CAP_{vk}$) of each product $k$ on ship $v$, the initial inventory level ($IS_{ik}$) for each product $k$ in harbor $i$, and the production/consumption rate ($R_{ik}$) of each product $k$ in harbor $i$. For each harbor $i$ that produces or consumes product $k$, this quantity is given by

$$m_i = \max_{k \in K_i^H} m_{ik},$$

where

$$m_{ik} = \begin{cases} \left\lceil \dfrac{T \times R_{ik} + (IS_{ik} - S_{MXik})}{\max_{v \in V}\{CAP_{vk}\}} \right\rceil & \text{if } J_{ik} = +1, \\ \left\lceil \dfrac{T \times R_{ik} + (S_{MNik} - IS_{ik})}{\max_{v \in V}\{CAP_{vk}\}} \right\rceil & \text{if } J_{ik} = -1 \end{cases}$$

is the minimum number of loadings (if $J_{ik} = +1$) or unloadings (if $J_{ik} = -1$) of product $k$ in harbor $i$ within the planning horizon.

The minimum (un)loadings $m_{ik}$ are determined based on the assumption that the ship with the largest capacity for product $k$ is the only one visiting harbor $i$. So we calculate how many times it needs to visit based on rate $R_{ik}$, stock level $IS_{ik}$, maximum harbor capacity $S_{MXik}$, and minimum harbor capacity $S_{MNik}$. For $J_{ik} = +1$, we assume that the vessel with the largest capacity for product $k$ loads at harbor $i$ when the inventory level is at $S_{MXik}$. Starting from level $IS_{ik}$, it takes $(S_{MXik} - IS_{ik})/R_{ik}$ time units for the storage tanks to reach this level. With

$$T - \left(\frac{S_{MXik} - IS_{ik}}{R_{ik}}\right) = \frac{TR_{ik} + (IS_{ik} - S_{MXik})}{R_{ik}}$$

time units remaining in the planning horizon, we again assume that the largest capacity vessel reloads when inventory has reached $S_{MXik}$. The time it takes to reach that level, starting from $S_{MXik} - \max_{v\in V}\{CAP_{vk}\}$, is

$$\frac{\max_{v\in V}\{CAP_{vk}\}}{R_{ik}}.$$

Therefore, after the first visit, this ship will need to reload

$$\frac{\frac{TR_{ik}+(IS_{ik}-S_{MXik})}{R_{ik}}}{\frac{\max_{v\in V}\{CAP_{vk}\}}{R_{ik}}} = \frac{TR_{ik} + (IS_{ik} - S_{MXik})}{\max_{v\in V}\{CAP_{vk}\}}$$

more times. By rounding up any fractional values for this quantity, we capture the first visit to yield the minimum number of visits $m_{ik}$ for product $k$ in harbor $i$. A similar argument holds for the case $J_{ik} = -1$ of harbors $i$ consuming product $k$, except now the largest capacity ship is unloading instead of loading.

To the minimum number of visits $m_i$, we add $m' \in \{1, 2, 3, 4\}$. Thus, each harbor has $m_i + m'$ positions. Also, for each harbor $i$, we fix the variable $y_{in} = 0$, $\forall n \leqslant m_i$, so that the harbor is visited at least $m_i$ times; otherwise, the problem would be infeasible.

Each test problem was solved four times by taking $m' \in \{1, 2, 3, 4\}$ in order to observe the impact on solution time of growth in the number of positions in the model. The results for the different settings were

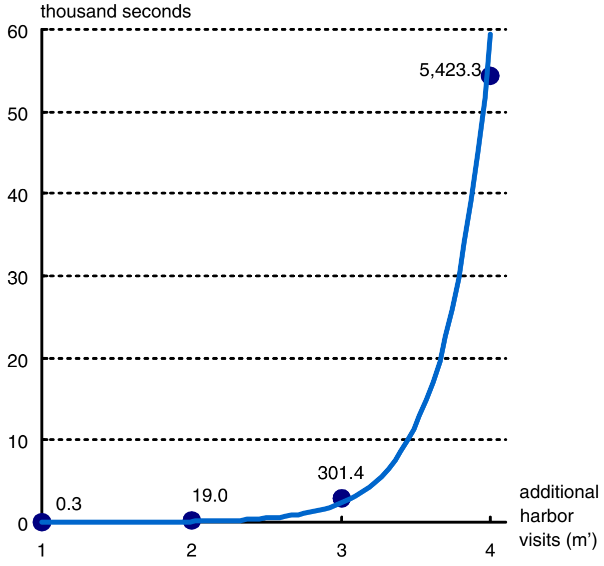


Fig. 5. Average solution times for ten $(|V|, |H_T|, |K|) = (3, 2, 2)$ test problems as function of $m'$.

Table 5
Average computing times (seconds) for each configuration

| Additional harbor visits | Configurations | | | | | |
|---|---|---|---|---|---|---|
| | (3, 2, 3) | (3, 3, 2) | (3, 3, 3) | (4, 2, 2) | (4, 2, 3) | (4, 3, 2) |
| 1 | 12.0 | 0.1 | 1.8 | 5.3 | 2.7 | 389.3 |
| 2 | 135.4 | 16.9 | 1535.3 | 205.8 | 491.5 | 9083.7 |
| 3 | 392.2 | 282.4 | 29579.4 | 1722.2 | 15812.4 | ≫15801.4$^{\dagger}$ |
| 4 | 953.0 | 2901.4 | 71675.8 | 14974.8 | ≫29328.4$^{*}$ | ≫67257.1$^{\ddagger}$ |

Table 6
Optimal costs and computing times (seconds) as number of possible harbor visits increases

| Additional harbor visits ($m'$) | Problems | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **1** (3, 2, 3) | | **2** (3, 2, 2) | | **3** (3, 3, 2) | | **4** (3, 3, 3) | |
| | Cost | Time | Cost | Time | Cost | Time | Cost | Time |
| 1 | inf | <0.01 | **36.2** | **0.1** | 70.9 | 0.4 | inf | <0.01 |
| 2 | **77.4** | **0.1** | 36.2 | 0.6 | **64.3** | **139.8** | **98.8** | **8938.7** |
| 3 | 77.4 | 0.5 | 36.2 | 21.9 | 64.3 | 2480.8 | **76.2** | **2.1E+5** |
| 4 | 77.4 | 2.1 | 36.2 | 371.5 | 64.3 | 25284.4 | **75.3** | **>4.5E+5** |

similar. The case $(|V|, |H_T|, |K|) = (3, 2, 2)$ is depicted in Fig. 5. The fitted curve was developed using exponential regression. The other cases are listed in Table 5 and show similar exponential growth characteristics. These results clearly suggest that it is advantageous to limit the number of possible positions in the model for each port. The asterisk (∗) indicates the average computing time for 4 problems, with 6 problems not terminating before the time limit of 4.5E+5 seconds. The dagger (†) indicates the average computing time for 4 problems, with 6 problems still running at time 4.5E+5 seconds. The double dagger (‡) indicates the average computing time for 2 problems, with 8 problems exceeding the time limit of 4.5E+5 seconds.

In the foregoing, we constructed different test problems by varying the number of possible visits $\mu_i = m_i + m'$ for $m' \in \{1, 2, 3, 4\}$. While solution times are faster for smaller $\mu_i$, it is possible for the problem to be infeasible if $\mu_i$ is too small for some $i$. This is illustrated in Table 6 which lists the results of test runs on four problems of different sizes taken from our one-hundred problem test bed.

For each problem, the objective cost and solution time is listed and the triple $(\cdot,\cdot,\cdot)$ denotes the problem settings $(|V|, |H_T|, |K|)$. While problems 2 and 3 only require $m' = 1$ to be feasible, that is not the case for problems 1 and 4. Moreover, problem 4 requires $m' \geqslant 4$ in order to find an optimal solution. We should point out that we are adding the same number of additional visits $m'$ to each harbor, so it is conceivable that the times for problem 4 can be reduced by allowing the additive amount to vary; i.e., by taking $\mu_i = m_i + m'_i$ for $m'_i \in \{1, 2, 3, 4\}$.

## 7. Concluding remarks

In this paper we have developed a comprehensive mathematical model for planning the sailing routes and loading/unloading schedules for a fleet of ships carrying liquid bulk products across a network of harbors during a specified planning horizon. The objective is to minimize the sum of the travel costs and the fixed costs incurred when products are loaded or unloaded. More precisely, our model is to optimize (O) subject to constraints (C1)–(C20). The model differs from existing work in this area in that it considers ships with multiple compartments that are dedicated to carrying different product types. Furthermore, our model allows the simultaneous servicing of multiple ships at a harbor, and requires loading and unloading to take place during inventory dependant time windows. We resolved some inherent nonlinearities in the problem by using some novel linearizing schemes from global optimization theory. We illustrated the model on a small example that was solved using a commercial solver for mixed integer linear programming. Numerical experiments with this solver demonstrate the need for specialized algorithms that exploit the structure inherent in the model. In particular, exponential growth in the solution times as the number of harbor visits increases is not surprising. However, the results in Table 6 suggest that a solution scheme that starts with a small number of possible visits and selectively increases this quantity should lead to a robust procedure that can solve larger problems than currently possible. This will be the topic of part II in the sequel.

## Acknowledgements

The authors are grateful to the anonymous referees for comments and recommendations that improved the presentation.

## Appendix A. Glossary of notation

### A.1. Variables

*Variables for network flows*
- $x_{imjnv}$: Arc flow variable is 1 if harbor arrivals $(i,m)$ and $(j,n)$ are directly connected in ship $v$'s route; otherwise, 0.
- $z_{imv}$: Route end indicator variable is 1 if $(i,m)$ is the end of the route for ship $v$; otherwise, 0.
- $y_{im}$: Slack variable is 1 if $(i,m)$ is not visited; otherwise, 0.

*Variables for loading and unloading*
- $l_{imvk}$: Load onboard in the compartment for product $k$ of ship $v$ when leaving $(i,m)$.
- $q_{imvk}$: Quantity of product $k$ loaded into or unloaded from ship $v$'s compartments in position $(i,m)$.

*Variables for time aspect*
- $o_{imvk}$: Binary variable is 1 if product $k$ is loaded or discharged at harbor arrival $(i,m)$ by ship $v$; otherwise, 0.
- $t_{Eim}$: Ending service time at $(i,m)$.

*Variables for inventories*
- $s_{imk}$: Stock level of product $k$ in harbor $i$ when service starts at $(i,m)$. Also we know the value of $s_{imk}$ for all $(i,m) \in S_F$.
- $s_{Eimk}$: Stock level of product $k$ in harbor $i$ when service finishes at $(i,m)$.

*Variables for stock levels*
- $p_{im}$: Binary variable is 0 of there are two or more ships at harbor $i$ during the $m$th arrival; otherwise, 0.

### A.2. Sets

*Sets for network flows*

- $S_T$: Set of all harbor arrivals $(i, m)$ for $i \in H_T$ and $m \in M_i$.
- $H_T$: Set of total harbors.
- $M_i$: Set of arrival numbers at harbor $i$.
- $S_0$: Set of initial positions $\{(i_v, m_v) \mid v \in V\}$. If more than one ship starts from the same harbor, then they are assigned a departure sequence number $m_v$; otherwise, $m_v = 1$.
- $V$: Set of available ships indexed by $v$.
- $H_v$: Set of harbors that can be visited by ship $v$.

*Sets for loading and unloading*

- $A_v$: Set of all feasible arcs for ship $v$.
- $K$: Set of products.
- $K_v$: Set of products that ship $v$ can carry.
- $K_i^H$: Set of products that harbor $i$ handles.

### A.3. Parameters

*Parameters for network flows*

- $i_v$: Starting harbor of vessel $v$.
- $m_v$: Assigned arrival sequence number for vessel $v$ in harbor $i_v$.

*Parameters for loading and unloading*

- $J_{jk}$: Indicator variable is 1 (if product $k$ is loaded at harbor $j$), 0 (if product $k$ passes through harbor $j$), or $-1$ (if product $k$ is unloaded at harbor $j$).
- $Q_{vk}$: Quantity of product $k$ on ship $v$ at start of planning horizon.
- $CAP_{vk}$: Capacity of the compartment for product $k$ in ship $v$.

*Parameters for time aspect*

- $TQ_{ik}$: Time required to load a unit of product $k$ at harbor $i$.
- $W_i$: Setup time to change products for loading and unloading at harbor $i$.
- $T_{ijv}$: Sailing time from harbor $i$ to harbor $j$.

*Parameters for inventories*

- $IS_{ik}$: Initial stock level of product $k$ at harbor $i$.
- $R_{ik}$: The consumption or production rate for product $k$ in harbor $i$.
- $S_{MNik}$: Minimum stock level at harbor $i$.
- $S_{MXik}$: Maximum stock level at harbor $i$.
- $T$: Length of planning period.

*Parameters for objective function*

- $C_{ijv}$: Cost for ship $v$ to sail from harbor $i$ to harbor $j$.
- $C_{Wik}$: Loading and unloading charges incurred at harbor $i$ for product $k$.

## Appendix B. General statement and proof of Proposition 2

Consider the nonlinear function

$$(f_1(x) - f_2(y))g(z)$$

over a domain such that each component function has known lower and upper bounds over its domain or subset of interest; i.e.,

$$L_{f_1} \leqslant f_1(x) \leqslant U_{f_1},$$
$$L_{f_2} \leqslant f_2(y) \leqslant U_{f_2},$$
$$L_g \leqslant g(z) \leqslant U_g.$$

Define $w = f_1(x) - f_2(y)$ which has bounds

$$L_{f_1} - U_{f_2} \leqslant w \leqslant U_{f_1} - L_{f_2}.$$

Let $u = wg(z)$, then

$$u \geqslant (L_{f_1} - U_{f_2})g(z) + L_g[f_1(x) - f_2(y)] - L_g(L_{f_1} - U_{f_2}) = \alpha,$$
$$u \geqslant (U_{f_1} - L_{f_2})g(z) + U_g[f_1(x) - f_2(y)] - U_g(U_{f_1} - L_{f_2}) = \beta,$$
$$u \leqslant (U_{f_1} - L_{f_2})g(z) + L_g[f_1(x) - f_2(y)] - L_g(U_{f_1} - L_{f_2}) = \gamma,$$
$$u \leqslant (L_{f_1} - U_{f_2})g(z) + U_g[f_1(x) - f_2(y)] - U_g(L_{f_1} - U_{f_2}) = \delta.$$

The latter four inequalities can be summarized as

$$\max\{\alpha, \beta\} \leqslant u \leqslant \min\{\gamma, \delta\}. \tag{12}$$

Now, let $v_1 = f_1(x)g(z)$ and $v_2 = f_2(y)g(z)$. Then

$$v_1 \geqslant L_g f_1(x) + L_{f_1} g(z) - L_{f_1} L_g = \alpha_1,$$
$$v_1 \geqslant U_{f_1} g(z) + U_g f_1(x) - U_{f_1} U_g = \beta_1,$$
$$v_1 \leqslant U_g f_1(x) + L_{f_1} g(z) - L_{f_1} U_g = \gamma_1,$$
$$v_1 \leqslant L_g f_1(x) + U_{f_1} g(z) - L_g U_{f_1} = \delta_1,$$
$$v_2 \geqslant L_g f_2(y) + L_{f_2} g(z) - L_{f_2} L_g = \alpha_2,$$
$$v_2 \geqslant U_{f_2} g(z) + U_g f_2(y) - U_{f_2} U_g = \beta_2,$$
$$v_2 \leqslant U_g f_2(y) + L_{f_2} g(z) - L_{f_2} U_g = \gamma_2,$$
$$v_2 \leqslant L_g f_2(y) + U_{f_2} g(z) - L_g U_{f_2} = \delta_2.$$

We can summarize the latter eight inequalities as

$$\max\{\alpha_1, \beta_1\} \leqslant v_1 \leqslant \min\{\gamma_1, \delta_1\}, \tag{13}$$
$$\max\{\alpha_2, \beta_2\} \leqslant v_2 \leqslant \min\{\gamma_2, \delta_2\}. \tag{14}$$

Since $u = v_1 - v_2$, the bounds on $v_1 - v_2$ can be determined from (13) and (14) as

$$\max\{\alpha_1, \beta_1\} - \min\{\gamma_2, \delta_2\} \leqslant v_1 - v_2 \leqslant \min\{\gamma_1, \delta_1\} - \max\{\alpha_2, \beta_2\}. \tag{15}$$

Moreover, since $\alpha = \alpha_1 - \delta_2$, $\beta = \beta_1 - \gamma_2$, $\gamma = \delta_1 - \alpha_2$, and $\delta = \gamma_1 - \beta_2$, the bounds given by (12) can be written as

$$\max\{\alpha_1 - \delta_2, \beta_1 - \gamma_2\} \leqslant v_1 - v_2 \leqslant \min\{\delta_1 - \alpha_2, \gamma_1 - \beta_2\}. \tag{16}$$

We will now show that the lower bounds specified by (15) are always greater than or equal to the lower bounds determined by (16) and the upper bounds of (15) are always less than or equal to the upper bounds of (16); i.e., the bounds on $v_1 - v_2$ given by (15) are tighter than those given by (16).

For the lower bound, there are two cases to consider.

*Case 1*. $\alpha_1 - \delta_2 \geqslant \beta_1 - \gamma_2$. Consider the four subcases

(i) $\alpha_1 \geqslant \beta_1, \gamma_2 \geqslant \delta_2$,
(ii) $\alpha_1 \leqslant \beta_1, \gamma_2 \geqslant \delta_2$,
(iii) $\alpha_1 \geqslant \beta_1, \gamma_2 \leqslant \delta_2$,
(iv) $\alpha_1 \leqslant \beta_1, \gamma_2 \leqslant \delta_2$.

For each of these subcases, it follows that

$$\max\{\alpha_1, \beta_1\} - \min\{\gamma_2, \delta_2\} \geqslant \alpha_1 - \delta_2 = \max\{\alpha_1 - \delta_2, \beta_1 - \gamma_2\}.$$

*Case 2*. $\beta_1 - \gamma_2 \geqslant \alpha_1 - \delta_2$. For the same foregoing subcases (i)–(iv), it follows that

$$\max\{\alpha_1, \beta_1\} - \min\{\gamma_2, \delta_2\} \geqslant \beta_1 - \gamma_2 = \max\{\alpha_1 - \delta_2, \beta_1 - \gamma_2\}.$$

An analogous argument is applied for the upper bound as follows.

*Case 1*. $\delta_1 - \alpha_2 \leqslant \gamma_1 - \beta_2$. Consider the four subcases

$\overline{\text{(i)}}$ $\delta_1 \geqslant \gamma_1,\ \alpha_2 \geqslant \beta_2$,
$\overline{\text{(ii)}}$ $\delta_1 \leqslant \gamma_1,\ \alpha_2 \geqslant \beta_2$,
$\overline{\text{(iii)}}$ $\delta_1 \geqslant \gamma_1,\ \alpha_2 \leqslant \beta_2$,
$\overline{\text{(iv)}}$ $\delta_1 \leqslant \gamma_1,\ \alpha_2 \leqslant \beta_2$.

For each of these subcases, it follows that

$$\min\{\gamma_1, \delta_1\} - \max\{\alpha_2, \beta_2\} \leqslant \delta_1 - \alpha_2 = \min\{\delta_1 - \alpha_2, \gamma_1 - \beta_2\}.$$

*Case 2*. $\gamma_1 - \beta_2 \leqslant \delta_1 - \alpha_2$. For the same foregoing subcases $\overline{\text{(i)}}$–$\overline{\text{(iv)}}$, it follows that

$$\min\{\gamma_1, \delta_1\} - \max\{\alpha_2, \beta_2\} \leqslant \delta_1 - \alpha_2 = \min\{\delta_1 - \alpha_2, \gamma_1 - \beta_2\}.$$

## References


[1] F. Al-Khayyal, Generalized bilinear programming: Part 1. Models, applications and linear programming relaxation, European Journal of Operational Research 60 (1992) 309–314.
[2] F. Al-Khayyal, J. Falk, Jointly constrained biconvex programming, Mathematics of Operations Research 8 (1983) 273–286.
[3] D.O. Bausch, G.G. Brown, D. Ronen, Scheduling short-term marine transport of bulk products, Maritime Policy and Management 25 (4) (1998) 335–348.
[4] G.G. Brown, G.W. Graves, D. Ronen, Scheduling ocean transportation of crude oil, Management Science 33 (3) (1987) 335–346.
[5] M. Christiansen, Decomposition of a combined inventory and time constrained ship routing problem, Transportation Science 33 (1) (1998) 3–16.
[6] M. Christiansen, K. Fagerholt, D. Ronen, Ship routing and scheduling: Status and perspectives, Transportation Science 38 (1) (2004) 1–18.
[7] M. Christiansen, B. Nygreen, A method for solving ship routing problems with inventory constraints, Annals of Operations Research 81 (1998) 357–378.
[8] M. Christiansen, B. Nygreen, Modeling path flows for a combined ship routing and inventory management problem, Annals of Operations Research 82 (1998) 391–421.
[9] K. Fagerholt, M. Christiansen, A combined ship scheduling and allocation problem, Journal of the Operational Research Society 51 (2000) 834–842.

[10] K. Fagerholt, M. Christiansen, A traveling salesman problem with allocation, time window and precedence constraints—an application to ship scheduling, International Transactions in Operational Research 7 (2000) 231–244.
[11] M. Fisher, Vehicle routing, in: M.O. Ball, T.L. Magnati, G.L. Nemhauser (Eds.), Handbooks in Operations Research and Management Science, Network Routing, vol. 8, Elsevier Science Publishers, Amsterdam, 1995, pp. 1–33.
[12] M. Fisher, M.B. Rosenwein, An interactive optimization system for bulk-cargo ship scheduling, Naval Research Logistics 36 (1989) 27–42.
[13] A.S. Jetlund, I.A. Karimi, Improving the logistics of multi-compartment chemical tankers, Computers and Chemical Engineering 28 (2004) 1267–1283.
[14] M. Miller, An interactive, computer-aided ship scheduling system, European Journal of Operational Research 32 (1987) 363–379.
[15] N.A. Papadakis, A.N. Perakis, A nonlinear approach to the multiorigin, multidestination fleet deployment problem, Naval Research Logistics 36 (1989) 515–528.
[16] H.N. Psaraftis, Forward to focused issue on maritime transportation, Transportation Science 33 (1) (1998).
[17] D. Ronen, Short-term scheduling of vessels for ship bulk or semi-bulk commodities originating in a single area, Operations Research 34 (1986) 164–173.
[18] H.D. Sherali, W.P. Adams, A Reformulation–Linearization Technique for Solving Discrete and Continuous Nonconvex Problems, Kluwer Academic, Boston, 1998.
[19] H.D. Sherali, A. Alameddine, An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes, Annals of Operations Research 25 (1990) 197–210.
[20] P. Toth, D. Vogo, The Vehicle Routing Problem, Society for Industrial and Applied Mathematic, 2002.