



Production, Manufacturing and Logistics

## An iterative two-phase hybrid matheuristic for a multi-product short sea inventory-routing problem

Ahmad Hemmati<sup>a</sup>, Lars Magnus Hvattum<sup>a,b,\*</sup>, Marielle Christiansen<sup>a</sup>, Gilbert Laporte<sup>c</sup><sup>a</sup> Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Alfred Getz veg 3, NO-7491, Trondheim, Norway<sup>b</sup> Faculty of Logistics, Molde University College, P.O. Box 2110, NO-6402, Molde, Norway<sup>c</sup> Canada Research Chair in Distribution Management, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine Montréal (Québec) Canada H3T 2A7

### ARTICLE INFO

#### Article history:

Received 2 February 2015

Accepted 29 January 2016

Available online 11 February 2016

#### Keywords:

Maritime transportation

Inventory-routing

Supply chain management

Adaptive large neighborhood search

Many-to-many distribution problem

### ABSTRACT

This paper considers a multi-product short sea inventory-routing problem in which a heterogeneous fleet of ships transports multiple products from production sites to consumption sites in a continuous time framework. A many-to-many distribution structure is taken into account, which makes it extremely hard to even compute feasible solutions. We propose an iterative two-phase hybrid matheuristic called Hybrid Cargo Generating and Routing (HCGR) to solve the problem. In the first phase the inventory-routing problem is converted into a ship routing and scheduling problem by generating cargoes subject to inventory limits through the use of mathematical programming. In the second phase, an adaptive large neighborhood search solves the resulting ship routing and scheduling problem. The HCGR heuristic iteratively modifies the generated cargoes based on information obtained during the process. The proposed heuristic is compared with an exact algorithm on small size instances; computational results are also presented on larger and more realistic instances.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Short sea shipping is one of the major modes of transportation for countries with a long shoreline or with many islands or rivers. In 2012, short sea shipping in the European Union (EU) countries accounted for nearly 1.8 billion tons of freight and for 60 percent of the total EU maritime transportation of goods (Eurostat, 2014). In short sea shipping, ships operate within a continent, make frequent port calls and often carry several products simultaneously. In contrast, in deep sea shipping, ships are much larger and sail between continents. The route structure for ships in deep sea operations is often simpler with pickup areas in one continent, and delivery areas in another. In short sea shipping, the sailing legs are relatively short compared to the time spent in ports. The port time consists of the sailing time in the inner port area, the

berthing time, the set up time required for loading and unloading the equipment, and the actual loading and unloading time.

In maritime transportation, there are often inventories located at both the loading and the unloading ports. When the actors involved in a maritime supply chain have the responsibility for both the transportation of the cargoes and the inventories at the ports, the underlying planning problem is called a maritime inventory-routing problem (MIRP) (Christiansen, Fagerholt, Nygreen, & Ronen, 2013). These inventory-routing problems are very complex. Because the shipping industry is capital intensive, a modest improvement in fleet utilization can translate into a large increase in profit. This means that there exists a great potential and need for research in solving real MIRP instances.

The problem studied in this paper is a multi-product short sea inventory-routing problem (SSIRP). Maritime inventory-routing problems (MIRPs) have attracted considerable attention in the last 15 years: see the surveys on MIRPs by Andersson, Hoff, Christiansen, Hasle, and Løkketangen (2010), Coelho, Cordeau, and Laporte (2014) and Papageorgiou, Nemhauser, Sokol, Cheo, and Keha (2014), and the recent general reviews on ship routing and scheduling by Christiansen and Fagerholt (2014) and Christiansen et al. (2013). Many of the MIRPs are concerned with the trans-

\* Corresponding author at: Faculty of Logistics, Molde University College, P.O. Box 2110, NO-6402, Molde, Norway. Tel.: +47 71214223.

E-mail addresses: [ahmad.hemmati@iot.ntnu.no](mailto:ahmad.hemmati@iot.ntnu.no) (A. Hemmati), [hvattum@himolde.no](mailto:hvattum@himolde.no), [lars.hvattum@iot.ntnu.no](mailto:lars.hvattum@iot.ntnu.no) (L.M. Hvattum), [mc@iot.ntnu.no](mailto:mc@iot.ntnu.no) (M. Christiansen), [gilbert.laporte@cirrelt.ca](mailto:gilbert.laporte@cirrelt.ca) (G. Laporte).

portation of a single product; see for instance the work by Agra, Andersson, Christiansen, and Wolsey (2013b), Christiansen (1999), Engineer, Furman, Nemhauser, Savelsbergh, and Song (2012), Fodstad et al. (2010), Furman, Song, Kocis, McDonald, and Warrick (2011), Goel, Furman, Song, and El-Bakry (2012), Grønhaug, Christiansen, Desaulniers, and Desrosiers (2010), Hewitt, Nemhauser, Savelsbergh, and Song (2013), Rocha et al. (2013), Shen, Chu, and Chen (2011), Sherali and Al-Yakoob, (2006a, 2006b), Song and Furman (2013) and Uggen et al. (2013). Ronen (2002) introduced a multi-product MIRP, but with no focus on ship routing. Several multi-product MIRP studies were later presented, such as those by Agra, Christiansen, and Delgado (2013a), Agra, Christiansen, Delgado, & Simonetti (2014), Al-Khayyal and Hwang (2007), Andersson (2011), Bilgen and Ozkaran (2007), Bredström, Carlsson, and Rönnqvist (2005), Christiansen et al. (2011), Dauzère-Pérès et al. (2007), Gunnarsson, Rönnqvist, and Carlsson (2006), Li, Karimi, and Srinivasan (2010), Persson and Göthe-Lundgren (2005), Rakke et al. (2011), Rakke, Andersson, Christiansen, & Desaulniers, 2013, Siswanto, Essam, and Sarker (2011), and Stålhane et al. (2012), where the transported products include pulp, wheat, cement, slurry, bitumen, fuel oil, liquefied natural gas products and general liquid bulk. The solution approaches include exact methods such as branch-and-cut and branch-and-price, and metaheuristics such as genetic algorithms and large neighborhood search. In addition, several algorithms are matheuristics, which combine mathematical programming and heuristics, e.g. rolling horizon heuristics and various fix-and-relax heuristics. Many of the published studies are based on real cases from the industry, in which it is important to meet the demand while taking the combined inventory management and routing decisions into account. As discussed in several papers (Andersson et al., 2010; Song & Furman, 2013), most known MIRPs are special versions of a basic problem, and tailor-made methods are necessary to solve the specific problems. In this paper, we consider a more general MIRP with several products and no particular geographical structure, and we develop an efficient heuristic to solve it.

Traditionally, maritime fleet routing and scheduling have been done manually by pencil and paper or by using simple spread sheets or information systems. The planning has to a large extent been based on the planners' knowledge and experience. Companies with combined inventory management and ship routing and scheduling issues have often handled the routing and scheduling components of the problem separately from inventory management to overcome the complex planning situation. In this way, a manual planner or simple inventory management system can produce a cargo plan that includes information about the specified quantities, pickup and delivery ports, and the time window for each cargo. Then a planner or ship routing and scheduling tool can take the specified cargoes as planning entities and design routing and scheduling plans. In Andersson et al. (2010) industrial aspects of combined inventory management and routing in maritime transportation are further discussed. These complex combined planning problems make it hard, if not impossible, to determine an optimal plan by manual planning methods or simple information systems. Therefore, the need for optimization-based decision support systems (DSSs) has increased and will probably continue to do so in the future. We have already witnessed the beginning of successful implementations of DSSs for MIRPs. Thus Dauzère-Pérès et al. (2007) describe a DSS for Omya Hustadmar-mor transporting calcium carbonate slurry. Furman et al. (2011) report upon the development of a DSS for feedstock routing in the ExxonMobil downstream sector. Finally, Fodstad et al. (2010) have developed a Liquefied Natural Gas (LNG) inventory routing model for Statoil and GDF SUEZ.

The contributions of this paper lie in (1) introducing a multi-product SSIRP with a many-to-many distribution structure that is

new to the literature; (2) creating realistically sized benchmark instances for the multi-product SSIRP with a many-to-many distribution structure; (3) designing a new effective and efficient iterative two-phase hybrid matheuristic, called the Hybrid Cargo Generating and Routing (HCGR) heuristic, capable of solving large and realistic instances of the multi-product SSIRP; (4) presenting a computational study which shows that the proposed heuristic can find high-quality solutions for large instances within short computing times, and including an analysis of the factors that influence the performance of the HCGR heuristic.

The remainder of this paper is organized as follows. The SSIRP is described in Section 2, together with a mathematical model of the problem. Section 3 presents the HCGR heuristic. Extensive computational results are reported and discussed in Section 4, followed by some concluding remarks in Section 5.

## 2. Problem description and mathematical model

In the multi-product SSIRP considered in this paper, each product can be produced and consumed in any number of ports. In a particular port, there may be a net production of one or several products and a net consumption of other products. It is possible to store the products in separate capacitated facilities at the ports. The net production and consumption rates are assumed to be constant. During the planning horizon the level of each inventory must lie within fixed lower and upper limits at all times. Separate limits apply to the end of the planning horizon and these may be tighter. It is assumed that the initial inventory storage levels are known at the beginning of the planning horizon. There is no pre-defined rule for the sequence of loading and unloading operations, thus the ships can load or unload the products in any order. It is assumed that there are lower and upper limits on the loaded and unloaded quantities, and that the associated handling times are quantity-dependent. These operations generate fixed and variable costs. The products are transported by a heterogeneous fleet, where ships differ in attributes such as capacity, speed, and transportation cost and time. A ship may load any types of products and no allocation of products to compartments onboard the ship is considered. Partial loading and unloading is allowed, and a ship can also handle different products at the same port visit. Two or more ships cannot load or unload simultaneously at a port. Start positions of the ships are given as inputs and the ships are assumed to be empty when they arrive at the first port, as well as at the end of their route.

The multi-product short sea inventory-routing problem consists of designing routes and schedules for the fleet in order to minimize the transportation and port costs, and to determine the quantities handled at each port call without exceeding the storage limits. To define the problem precisely, we introduce a continuous time arc-flow formulation of the problem with the following notation. Throughout the paper, we use lower case letters for variables and upper case letters for parameters and sets.

### Indices

$i, j$ :	ports
$v$ :	ships
$k$ :	products
$m, n$ :	visit numbers in a port
$(i, m)$ :	$m^{\text{th}}$ visit to port $i$
$(i, m, j, n)$ :	direct ship movements (arcs) from port visit $(i, m)$ to port visit $(j, n)$

### Sets

$N$ :	set of ports
$V$ :	set of ships
$K_v$ :	set of products $k$ that ship $v$ can transport
$S_v^A$ :	set of possible port visits $(i, m)$ for ship $v$

$S_v^X$ : set of all possible movements  $(i, m, j, n)$  for ship  $v$

#### Parameters

$J_{ik}$ : 1 if port  $i$  is a producer of product  $k$ ;  $-1$  if port  $i$  is a consumer of product  $k$ , and 0 if  $i$  is neither a producer nor a consumer of product  $k$

$C_v$ : capacity of ship  $v$

$\underline{Q}_{ik}, \bar{Q}_{ik}$ : minimum and maximum limits on the quantity of product  $k$  that may be unloaded or loaded at port  $i$  at each visit

$T_{ik}^Q$ : time to unload or load one unit of product  $k$  at port  $i$

$T_{iv}^O$ : transportation time from ship  $v$ 's initial position to port  $i$

$T_{ijv}$ : transportation time between ports  $i$  and  $j$  including setup time in ports by ship  $v$

$\bar{T}$ : length of the time horizon

$\bar{\mu}_i$ : upper bound on the number of visits to port  $i$

$S_{ik}^O$ : initial stock level of product  $k$  at port  $i$

$R_{ik}$ : net production or consumption rate of product  $k$  at port  $i$

$\underline{S}_{ik}, \bar{S}_{ik}$ : minimum and maximum stock levels (tank capacity) of product  $k$  at port  $i$

$\underline{S}_{ik}^T, \bar{S}_{ik}^T$ : minimum and maximum stock levels of product  $k$  at port  $i$  at the end of planning horizon

$C_{ijv}^T$ : total transportation cost from ports  $i$  and  $j$  with ship  $v$

$C_{iv}^{TO}$ : transportation cost from ship  $v$ 's origin to port  $i$

$C_{ik}^O$ : fixed cost of handling product  $k$  at port  $i$

$C_{ik}^Q$ : unloading or loading cost per unit of product  $k$  at port  $i$

#### Variables

$x_{imjnv}$ : 1 if ship  $v$  sails directly from port visit  $(i, m)$  to port visit  $(j, n)$  and 0 otherwise

$x_{imv}^O$ : 1 if ship  $v$  sails directly from its initial position to port visit  $(i, m)$  and 0 otherwise

$z_{imv}$ : 1 if ship  $v$  ends its route at port visit  $(i, m)$  and 0 otherwise

$z_v^O$ : 1 if ship  $v$  ends its route at the origin (it is not used) and 0 otherwise

$w_{imv}$ : 1 if ship  $v$  arrives port  $i$  at visit number  $m$  and 0 otherwise

$y_{im}$ : 1 if a ship arrives port  $i$  at visit number  $m$  and 0 otherwise

$o_{imvk}$ : 1 if product  $k$  is loaded onto or unloaded from ship  $v$  at port visit  $(i, m)$  and 0 otherwise

$l_{imvk}$ : amount of product  $k$  on board ship  $v$  when leaving from port visit  $(i, m)$

$q_{imvk}$ : amount of product  $k$  unloaded or loaded at port visit  $(i, m)$  by ship  $v$

$t_{im}$ : start time of unloading or loading at port visit  $(i, m)$

$t_{im}^E$ : end time of unloading or loading at port visit  $(i, m)$

$s_{imk}$ : inventory levels of product  $k$  at the start of unloading or loading at port visit  $(i, m)$

$s_{imk}^E$ : inventory levels of product  $k$  at the end of unloading or loading at port visit  $(i, m)$

#### Objective function

$$\begin{aligned} \text{Minimize } & \sum_{v \in V} \sum_{(i, m, j, n) \in S_v^X} C_{ijv}^T \cdot x_{imjnv} + \sum_{v \in V} \sum_{(i, m) \in S_v^A} C_{iv}^{TO} \cdot x_{imv}^O \\ & + \sum_{v \in V} \sum_{(i, m) \in S_v^A} \sum_{k \in K_v} (C_{ik}^O \cdot o_{imvk} + C_{ik}^Q \cdot q_{imvk}) \end{aligned} \quad (1)$$

subject to

#### Routing constraints

$$\sum_{(i, m) \in S_v^A} x_{imv}^O + z_v^O = 1 \quad v \in V \quad (2)$$

$$x_{imv}^O + \sum_{(j, n, i, m) \in S_v^X} x_{jnimv} - w_{imv} = 0 \quad v \in V, (i, m) \in S_v^A \quad (3)$$

$$w_{imv} - \sum_{(i, m, j, n) \in S_v^X} x_{imjnv} - z_{imv} = 0 \quad v \in V, (i, m) \in S_v^A \quad (4)$$

$$\sum_{v \in V} w_{imv} = y_{im} \quad (i, m) \in S^A \quad (5)$$

$$y_{i(m-1)} - y_{im} \geq 0 \quad (i, m) \in S^A : m \geq 2 \quad (6)$$

$$x_{imv}^O, w_{imv}, z_{imv} \in \{0, 1\} \quad v \in V, (i, m) \in S_v^A \quad (7)$$

$$x_{imjnv} \in \{0, 1\} \quad v \in V, (i, m, j, n) \in S_v^X \quad (8)$$

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

$$z_v^O \in \{0, 1\} \quad v \in V \quad (10)$$

#### Loading and unloading constraints

$$x_{imjnv} \cdot (l_{imvk} + J_{jk} q_{jnvk} - l_{jnvk}) = 0 \quad v \in V, (i, m, j, n) \in S_v^X, k \in K_v \quad (11)$$

$$x_{imv}^O \cdot (J_{ik} q_{imvk} - l_{imvk}) = 0 \quad v \in V, (i, m) \in S_v^A, k \in K_v \quad (12)$$

$$\sum_{k \in K_v} l_{imvk} \leq C_v \cdot \sum_{(j, n) \in S_v^A} x_{imjnv} \quad v \in V, (i, m) \in S_v^A \quad (13)$$

$$\underline{Q}_{ik} \cdot o_{imvk} \leq q_{imvk} \leq \bar{Q}_{ik} \cdot o_{imvk} \quad v \in V, (i, m) \in S_v^A, k \in K_v \quad (14)$$

$$\sum_{k \in K_v} o_{imvk} \geq w_{imv} \quad v \in V, (i, m) \in S_v^A \quad (15)$$

$$o_{imvk} \leq w_{imv} \quad v \in V, (i, m) \in S_v^A, k \in K_v \quad (16)$$

$$l_{imvk}, q_{imvk} \geq 0 \quad v \in V, (i, m) \in S_v^A, k \in K_v \quad (17)$$

$$o_{imvk} \in \{0, 1\} \quad v \in V, (i, m) \in S_v^A, k \in K_v \quad (18)$$

#### Time constraints

$$t_{im}^E \geq t_{im} + \sum_{v \in V} \sum_{k \in K_v} T_{ik}^Q \cdot q_{imvk} \quad (i, m) \in S^A \quad (19)$$

$$t_{im} - t_{i(m-1)}^E \geq 0 \quad (i, m) \in S^A : m \geq 2 \quad (20)$$

$$x_{imjnv} \cdot (t_{im}^E + T_{ijv} - t_{jn}) \leq 0 \quad v \in V, (i, m, j, n) \in S_v^X \quad (21)$$

$$\sum_{v \in V} T_{iv}^O \cdot x_{imv}^O \leq t_{im} \quad (i, m) \in S^A \quad (22)$$

$$t_{im}, t_{im}^E \geq 0 \quad (i, m) \in S^A \quad (23)$$

#### Inventory constraints

$$s_{i1k} = S_{ik}^O + J_{ik} \cdot R_{ik} \cdot t_{i1} \quad i \in N, k \in K \quad (24)$$

$$s_{imk}^E = s_{imk} - \sum_{v \in V} J_{ik} \cdot q_{imvk} + J_{ik} \cdot R_{ik} \cdot (t_{im}^E - t_{im}) \quad (i, m) \in S^A, k \in K \quad (25)$$

$$s_{imk} = s_{i(m-1)k}^E + J_{ik} \cdot R_{ik} (t_{im} - t_{i(m-1)}^E) \quad (i, m) \in S^A : m \geq 2, k \in K \quad (26)$$

$$\underline{S}_{ik} \leq s_{imk}, s_{imk}^E \leq \bar{S}_{ik} \quad (i, m) \in S^A, k \in K \quad (27)$$

$$\underline{S}_{ik}^T \leq \underline{S}_{i\bar{\mu}_i,k}^E + J_{ik} \cdot R_{ik} \cdot (\bar{T} - t_{i\bar{\mu}_i}^E) \leq \bar{S}_{ik}^T \quad i \in N, k \in K \quad (28)$$

$$s_{imk}, s_{imk}^E \geq 0 \quad (i, m) \in S^A, k \in K. \quad (29)$$

In this formulation, the objective (1) is to minimize the total transportation and port operating cost. Eq. (2) ensure that each ship departs from its initial position and sails towards another port, or is not used. Eqs. (3) and (4) are the flow conservation constraints, ensuring that a ship arriving at a port also leaves that port either by visiting another port or ends its route. Constraints (5) ensure that each port visit  $(i, m)$  is made at most once. Constraints (6) state that if port  $i$  is visited  $m$  times, then it must also have been visited  $m - 1$  times before. Constraints (7)–(10) define the variables as binary.

The non-linear equations (11) determine the quantity of product  $k$  on board ship  $\nu$  when it sails from port visit  $(i, m)$  to port visit  $(j, n)$ . Constraints (12) are similar to (11) and determine the load on board the ship at the first visit. The ship capacities are enforced through constraints (13). Constraints (14) impose lower and upper limits on the loaded and unloaded quantities. Constraints (15) ensure that if ship  $\nu$  makes port visit  $(i, m)$ , then at least one product must be unloaded or loaded. Constraints (16) ensure that if ship  $\nu$  unloads or loads one product at visit  $(i, m)$ , then  $w_{im\nu}$  must be equal to one. Constraints (17)–(18) define the domain of the variables.

Constraints (19) define the end time of service of port visit  $(i, m)$ . Constraints (20) state that if port visit  $(i, m)$  starts, then port visit  $(i, m - 1)$  must have ended no later than the start of visit  $(i, m)$ . The non-linear constraints (21) relate the end time of port visit  $(i, m)$  to the start time of port visit  $(j, n)$  when ship  $\nu$  sails directly from  $(i, m)$  to  $(j, n)$ . Constraints (22) ensure that if ship  $\nu$  travels from its initial position directly to port visit  $(i, m)$ , then the start time is at least the transportation time between these two positions. Constraints (23) define the continuous time variables. Constraints (11) and (21) are converted to linear constraints similarly to what was done in [Agra et al. \(2013a\)](#).

Eq. (24) provides the inventory level of each product at the first visit. Eq. (25) calculates the inventory level of each product when the service ends at port visit  $(i, m)$ . Similarly, Eq. (26) relates the inventory level at the start of port visit  $(i, m)$  to the inventory level at the end of port visit  $(i, m - 1)$ . The upper and lower bounds on the inventory levels are ensured by constraints (27), and constraints (28) are imposed to deal with the stock levels at the end of the planning horizon. Finally the non-negativity of the inventory level variables is enforced by constraints (29).

### 3. Hybrid cargo generating and routing heuristic

The HCGR heuristic we have developed for the SSIRP is based on the idea of [Hemmati, Stålhane, Hvattum, and Andersson \(2015\)](#) to convert an inventory-routing problem into a ship routing and scheduling problem. In the ship routing and scheduling problem, a shipping company has a set of cargoes that it is committed to carry. Each cargo must be picked up at its port of loading, transported, and delivered to its corresponding unloading port. Time windows are specified for the allowed arrival time in ports. Cargoes with specified origin and destination ports, a quantity to transport, and time windows for both the origin and the destination ports, should therefore be defined in order to convert the SSIRP into a ship routing and scheduling problem. [Hemmati et al. \(2015\)](#) introduced a heuristic to solve a combined cargo routing and inventory-routing problem faced by a shipping company that offers vendor-managed inventory services to some of its customers. They assumed that the inventories are paired, and then introduced cargoes to replace the inventory management for each inventory pair. In this paper, instead of paired inventories, there are invento-

ries for multiple products with a many-to-many distribution structure, which makes the procedure of defining the representative cargoes more difficult. Consequently, the main steps of our algorithm that define the cargoes are different from those of their algorithm.

To accomplish the required conversion, we transform the inventory management aspect of the problem into a set of cargoes with known quantities and time windows based on the information from inventory levels, the inventory limits, and the production or consumption rates. Thus, in the first phase, the SSIRP is converted into a ship routing and scheduling problem by generating cargoes based on the inventory information. The loaded and unloaded quantities for each of the cargoes generated are determined by solving a transportation model, and the time windows for start of unloading or loading operations are obtained by solving a time window model. This model generates wide, feasible time windows at the origin and destination ports corresponding to each cargo. The ship routing and scheduling problem is then solved by means of an adaptive large neighborhood search (ALNS) heuristic, a solution framework introduced by [Ropke and Pisinger \(2006\)](#). Furthermore, the solution is analyzed by means of several criteria and the HCGR heuristic iteratively changes the generated cargoes based on information obtained during this process.

The HCGR first computes the minimum total quantity of each product that should be loaded or unloaded at each port, considering the net production or consumption rate, the stock limits, and the stock levels at the beginning of the planning horizon. The transportation model is then applied, given the total quantities and optimistic transportation costs. The purpose of the model is to determine the amount of each product to transfer from each origin port to each destination port. Such a transfer corresponds to a cargo. Since there are limits on the quantity unloaded or loaded at ports, several cargoes may be needed to transport a given product from a specific origin to a specific destination. These cargoes will be referred to as a homogeneous cargo set, that is, a set of cargoes that have the same origin and destination ports but possibly representing different amounts of cargo and having different time windows. Cargoes belonging to a homogeneous cargo set are eventually changed in the HCGR heuristic to iteratively define new ship routing and scheduling problems. We refer to the remaining cargoes that do not belong to any homogeneous cargo set as ordinary cargoes. These do not change during the execution of the HCGR heuristic.

In the ship routing and scheduling problem, each cargo has separate time windows for the loading port and the unloading port. In the next step, the HCGR considers all inventory limits for each product in order to calculate valid time windows for the cargoes at their loading and unloading ports. The model guarantees that the inventory levels will remain within their limits as long as the cargoes are loaded and unloaded within these windows.

Having specified the initial cargoes and time windows, a first ship routing and scheduling problem is solved by an ALNS heuristic. After the first ship routing and scheduling problem has been solved, the HCGR heuristic creates new alternative cargo quantities and adds them to a list. The new cargo quantities are defined for each homogeneous cargo set based on criteria presented in [Section 3.6](#). The HCGR heuristic then generates all possible combinations of these cargo quantities over all the homogeneous cargo sets. The number of these combinations grows exponentially in parameters such as the number of inventories, the number of products, and the length of the planning horizon. To reduce solution times, the ALNS is not applied to all possible cargo combinations. Instead, we apply a  $k$ -means clustering algorithm to partition the combinations into clusters of similar combinations. An iterative process is then used to select new cargo quantity combinations that are used as input to the ALNS.



### 3.1. Overview of the HCGR heuristic

The pseudo-code of the algorithm is presented in Algorithm 1.

### 3.2. Determining total quantities to transport between ports

In line 4, the algorithm determines, for each product  $k$  and each port  $i$ , the minimum total quantity,  $Q_{ik}^T$ , that should be loaded or unloaded to ensure that the inventory level for product  $k$  at port  $i$  will remain within its limits at the end of the planning horizon. The minimum total quantities are computed according to Eq. (30) for the production sites ( $J_{ik} > 0$ ) and the consumption sites ( $J_{ik} < 0$ ).

$$Q_{ik}^T = \begin{cases} S_{ik}^O + (\bar{T} \cdot R_{ik}) - \bar{S}_{ik}^T, & J_{ik} > 0 \\ \bar{S}_{ik}^T + (\bar{T} \cdot R_{ik}) - S_{ik}^O, & J_{ik} < 0. \end{cases} \quad (30)$$

### 3.3. Transportation model

Having determined the quantities  $Q_{ik}^T$  that must be handled at each port for each product, the next step is to determine, for each product, how many times cargo should be transported between each pair of production and consumption ports. To this end, we define a modified transportation model with limits on the amounts that can be transported. These limits are derived from instance-specific data, with given lower and upper limits on the quantities loaded and unloaded. The modified transportation model minimizes the total transportation and cargo handling costs while considering the minimum total quantity  $Q_{ik}^T$  loaded at each production site, the minimum total quantity  $Q_{jk}^T$  unloaded at each consumption site, and upper and lower limits on the amounts  $\bar{Q}_{ijk}$  and  $\underline{Q}_{ijk}$  transported. The model is applied to each product  $k$  separately. The additional notation introduced for this model is given below:

$h$ :	transportation cargo number between two specific ports
$C_{ij}$ :	minimum transportation cost from port $i$ to $j$ ( $C_{ij} = \min_{v \in V} C_{ijv}^T$ , $i, j \in N$ )
$H$ :	set of transportation cargoes between two specific ports
$\bar{\mu}_{ik}$ :	upper bound on the number of visits to port $i$ for product $k$
$\bar{Q}_{ijk}$ :	upper limit on the quantity of product $k$ transporting from port $i$ to $j$ , $\bar{Q}_{ijk} = \min(\bar{Q}_{ik}, \bar{Q}_{jk}, \min_v C_v)$
$\underline{Q}_{ijk}$ :	lower limit on the quantity of product $k$ transporting from port $i$ to $j$ , $\underline{Q}_{ijk} = \max(Q_{ik}, Q_{jk})$
$x_{ijh}$ :	quantity transported from port $i$ to port $j$ with cargo $h = 1$ if the $h^{th}$ cargo from port $i$ to port $j$ exists, and 0 otherwise
$y_{ijh}$ :	
$\varepsilon_{ij}$ and $\delta_{ij}$ :	perturbations of the objective coefficients for cargoes from port $i$ to port $j$

The index  $h$ , the set  $H$ , and the decision variables  $x_{ijh}$  and  $y_{ijh}$  are defined for product  $k$ . The transportation model for product  $k$  can be stated as follows:

$$\text{Minimize } \sum_{i \in N} \sum_{j \in N} \sum_{h \in H} (C_{ij} + C_{ik}^O + C_{jk}^O + \varepsilon_{ij}) \cdot y_{ijh} + (C_{ik}^O + C_{jk}^O + \delta_{ij}) \cdot x_{ijh} \quad (31)$$

subject to

$$\sum_{j \in N} \sum_{h \in H} x_{ijh} \geq Q_{ik}^T \quad i \in N, J_{ik} > 0 \quad (32)$$

$$\sum_{i \in N} \sum_{h \in H} x_{ijh} \geq Q_{jk}^T \quad j \in N, J_{jk} < 0 \quad (33)$$

$$\underline{Q}_{ijk} \cdot y_{ijh} \leq x_{ijh} \leq \bar{Q}_{ijk} \cdot y_{ijh} \quad i, j \in N, h \in H, J_{ik} > 0, J_{jk} < 0 \quad (34)$$

$$\sum_{j \in N} \sum_{h \in H} y_{ijh} \leq \bar{\mu}_{ik} \quad i \in N, h \in H, J_{ik} > 0 \quad (35)$$

$$\sum_{i \in N} \sum_{h \in H} y_{ijh} \leq \bar{\mu}_{jk} \quad j \in N, h \in H, J_{jk} < 0 \quad (36)$$

$$y_{ij(h-1)} - y_{ijh} \geq 0 \quad i, j \in N, h \in H \setminus \{1\}, J_{ik} > 0, J_{jk} < 0 \quad (37)$$

$$x_{ijh} \geq 0 \quad i, j \in N, h \in H, J_{ik} > 0, J_{jk} < 0 \quad (38)$$

$$y_{ijh} \in \{0, 1\} \quad i, j \in N, h \in H, J_{ik} > 0, J_{jk} < 0. \quad (39)$$

The objective (31) minimizes the sum of the total transportation cost, the fixed operating cost at the ports, and the unit loading and unloading costs at the ports. Perturbations may be added to the objective function, giving the possibility of obtaining different solutions. Constraints (32) ensure that the minimum required quantity is loaded over the planning horizon at ports with net production, and constraints (33) ensure that the minimum required quantity is unloaded at ports with net consumption. Constraints (34) impose lower and upper limits on the transportation amounts, while constraints (35) and (36) control the number of visits at the ports. Constraints (37) state that if cargo  $h$  for product  $k$  from port  $i$  to  $j$  exists, with  $h > 1$ , then cargo  $h - 1$  must also exist. Constraints (38) and (39) are non-negativity and integrality constraints. In the solution of the model, if  $x_{ijh} > 0$  then a cargo is defined from port  $i$  to  $j$  with a quantity equal to  $x_{ijh}$ .

In our proposed HCGR heuristic we set  $\bar{\mu}_{ik}$  equal to  $\bar{\mu}_i$ . If the transportation model does not find a feasible solution for all products, the HCGR heuristic ends without any feasible solution to the overall problem. However, the main problem may in some cases have a feasible solution, since the upper limit on the quantity of product  $k$  transported from port  $i$  to  $j$  in constraints (34) can be tighter than the original upper limit in constraints (14).

### 3.4. Time window model

Each time a set of cargoes for product  $k$  is generated, we use a time window model to generate wide feasible time windows for the origin and destination ports corresponding to each cargo. This model guarantees that the inventory levels for product  $k$  remain within their limits throughout the planning horizon as long as the cargoes are loaded and unloaded within the time windows.

Here we provide the new notation for the time window model:

$G_k$ :	set of cargoes for product $k$ indexed by $g$
$P_g$ :	loading port of the cargo $g$
$D_g$ :	unloading port of the cargo $g$
$Q_g$ :	quantity of the cargo $g$
$S_{ig}^S$ :	cumulative sum of the cargo quantities unloaded or loaded from or to port $i$ just before unloading or loading cargo $g$ at port $i$
$S_{ig}^E$ :	cumulative sum of the cargo quantities unloaded or loaded from or to port $i$ just after unloading or loading cargo $g$ at port $i$ ( $S_{ig}^E = S_{ig}^S + Q_g$ )
$T_{ij}^M$ :	minimum transportation time from port $i$ to port $j$ ( $T_{ij}^M = \min_{v \in V} \{T_{ijv}\}$ , $i, j \in N$ )

$[t_g^{PS}, t_g^{PF}]$ : time window for cargo  $g$  at loading port

$[t_g^{DS}, t_g^{DF}]$ : time window for cargo  $g$  at unloading port

We can now state the mathematical formulation for the time window, for each  $k \in K$  and  $g \in G_k$ :

$$\text{Maximize } t_g^{PF} - t_g^{PS} + t_g^{DF} - t_g^{DS}, \quad (40)$$

subject to

$$S_{P_gk}^O \leq S_{P_gk}^S - S_{P_gk}^S + (R_{P_gk} \cdot t_g^{PS}) \leq \bar{S}_{P_gk} \quad (41)$$

**Algorithm 1** HCGR heuristic.

---

```

1:  Inputs: time horizon, number of ports, number of ships, number of products, net production-consumption rates, lower stock limits, upper stock limits,
    initial stocks, ending lower and upper stock limits, ship capacities, ship-port compatibility, transportation time table, transportation cost table, minimum
    and maximum limits on the loaded and unloaded quantity, loading and unloading times per unit, fixed costs, costs per unit
2:  For each product  $k$ 
3:    For each port  $i$ 
4:      Determining total quantity of each product to be transported from and to each port (Section 3.2)
5:    End-for
6:    Solve the Transportation Model to initialize the cargoes for product  $k$  with quantities, origins and destinations (Section 3.3)
7:    Solve the Time Window Model to find the time windows for the cargoes (Section 3.4)
8:    Find the fixed and per unit unloading or loading times per cargo
9:  End-for
10: Solve the cargo routing problem by ALNS considering all cargoes to get a solution,  $S$ , and an objective function value,  $f(S)$  (Section 3.5)
11: Find homogeneous cargo set
12: For each homogeneous cargo set
13:   For each cargo inside the set
14:     Create pool of New cargo quantity (Section 3.6)
15:   End-for
16: End-for
17: Generate all possible combinations of cargo quantities
18: Use  $k$ -means clustering algorithm to cluster the combinations (Section 3.7)
19: For (Iteration < Max Iteration)
20:   Cluster Selection to select potentially improving clusters (Section 3.8)
21:   Pick a random combination from the selected cluster which is not explored yet
22:   Update the quantities of the cargoes using the combination picked
23:   Optimize the Time Window Model to find the time windows for new cargoes (Section 3.4)
24:   Find the fixed and per unit loading and unloading times per cargo based on new quantities
25:   Solve the new cargo routing problem by ALNS considering all cargoes and get the solution,  $S'$ , and its objective function value,  $f(S')$  (Section 3.5)
26:   If ( $f(S') < f(S)$ ) then
27:      $S = S'$ 
28:   End-if
29:   adjust cluster weights (Section 3.8)
30: End-for

```

---

$$\underline{S}_{p_gk} \leq S_{p_gk}^O - S_{p_gk}^S + (R_{p_gk} \cdot t_g^{PF}) \leq \bar{S}_{p_gk} \quad (42)$$

$$\underline{S}_{D_gk} \leq S_{D_gk}^O + S_{D_gk}^S - (R_{D_gk} \cdot t_g^{DS}) \leq \bar{S}_{D_gk} \quad (43)$$

$$\underline{S}_{D_gk} \leq S_{D_gk}^O + S_{D_gk}^S - (R_{D_gk} \cdot t_g^{DF}) \leq \bar{S}_{D_gk} \quad (44)$$

$$\underline{S}_{p_gk} \leq S_{p_gk}^O - S_{p_gk}^E + (R_{p_gk} \cdot T_{p_gk}^Q \cdot Q_g) + (R_{p_gk} \cdot t_g^{PS}) \leq \bar{S}_{p_gk} \quad (45)$$

$$\underline{S}_{p_gk} \leq S_{p_gk}^O - S_{p_gk}^E + (R_{p_gk} \cdot T_{p_gk}^Q \cdot Q_g) + (R_{p_gk} \cdot t_g^{PF}) \leq \bar{S}_{p_gk} \quad (46)$$

$$\underline{S}_{D_gk} \leq S_{D_gk}^O + S_{D_gk}^E - (R_{D_gk} \cdot T_{D_gk}^Q \cdot Q_g) - (R_{D_gk} \cdot t_g^{DS}) \leq \bar{S}_{D_gk} \quad (47)$$

$$\underline{S}_{D_gk} \leq S_{D_gk}^O + S_{D_gk}^E - (R_{D_gk} \cdot T_{D_gk}^Q \cdot Q_g) - (R_{D_gk} \cdot t_g^{DF}) \leq \bar{S}_{D_gk} \quad (48)$$

$$t_g^{PS} \leq t_g^{PF} \quad (49)$$

$$t_g^{DS} \leq t_g^{DF} \quad (50)$$

$$t_g^{PF} \leq \bar{T} \quad (51)$$

$$t_g^{DF} \leq \bar{T} \quad (52)$$

$$t_g^{DS} - t_g^{PS} \geq T_{p_gD_g}^M \quad (53)$$

$$t_g^{DF} - t_g^{PF} \geq T_{p_gD_g}^M \quad (54)$$

$$t_g^{PS}, t_g^{PF}, t_g^{DS}, t_g^{DF} \geq 0. \quad (55)$$

(49) and (50) state that the time window must start before it ends and constraints (51) and (52) state that time window must end before the end of planning horizon. Constraints (53)–(54) mean that the time window at the unloading node may not start or end earlier than the start or end time at the loading node plus the shortest transportation time between them. Finally, constraints (55) express the non-negativity requirements for the variables.

### 3.5. Adaptive large neighborhood search

The ALNS heuristic is used to solve the ship routing and scheduling problem generated by defining cargoes and corresponding time windows. A solution is represented by one vector for each of the actual ships, and one for an artificial ship that contains all cargoes not transported by the actual ships. Each vector consists of a visiting sequence of ports representing the order in which they are visited along the route of the ship. The proposed ALNS starts with a solution in which all cargoes are serviced by the artificial ship. This solution is infeasible for the SSIRP, but the ALNS considers it as a feasible solution for the ship routing and scheduling problem. Thus, there is no sailing cost for the initial solution, and the only cost is the cost of not transporting the cargoes. These costs are set to very high values, thereby pushing the algorithm towards transporting all the cargoes by the actual ships. At each iteration of the main algorithm, a new set of cargoes is considered by the ALNS heuristic. The main algorithm fails if it ends in a solution that uses the artificial ship. This means that the main algorithm could not find a set of cargoes that can be transported only by the actual ships.

The ALNS heuristic uses several competing removal and insertion operators. At each iteration, using statistics gathered during the search, the ALNS heuristic chooses one removal operators to remove cargoes from the solution and one insertion operator to re-insert the cargoes in the solution. Because this type of algorithm works with several operators, it is very well suited to problems such as ours in which the range of potential decisions is

The objective function (40) maximizes the width of the time windows, while constraints (41)–(48) ensure that the inventories stay within their limits during the planning horizon. Constraints

**Algorithm 2** ALNS heuristic for a ship routing and scheduling problem.

---

```

1: Function ALNS
2:   generate initial solution  $s$  with objective function of  $f(s)$ 
3:    $S_{best} = s$ ,  $f(S_{best}) = f(s)$ 
4:   Repeat
5:      $s' = s$ 
6:     select removal and insertion operators based on search parameters
7:     choose the number,  $q$ , of cargoes to remove
8:     remove  $q$  cargoes from  $s'$ 
9:     reinsert the removed cargoes into  $s'$ 
10:    If ( $f(s') < f(S_{best})$ ) then
11:       $S_{best} = s'$ 
12:    End-if
13:    If accept ( $s'$ ,  $s$ ) then
14:       $s = s'$ 
15:    End-if
16:    update search parameters
17:  Until stop-criterion met
18:  return  $S_{best}$ 

```

---

very broad. The general pseudo-code for our ALNS implementation is based on the heuristic of Hemmati, Hvattum, Fagerholt, and Norstand (2014) for the cargo routing problem. It is described in Algorithm 2:

The ALNS heuristic starts with the initial solution. At each iteration,  $q$  cargoes are removed from the current solution and then reinserted in new positions. The value of  $q$  is chosen randomly from the interval  $[4, \min(100, \xi n)]$  where  $n$  is the number of cargoes and  $\xi$  is a constant parameter. Thus, the number of removed cargoes varies during the search to provide different neighborhood sizes. The algorithm uses three different removal operators: the Shaw removal operator, a random-removal operator, and a worst-removal operator. The Shaw removal operator is based on removing a set of similar cargoes, while the random-removal operator selects  $q$  cargoes at random and removes them from the solution. Finally, the worst-removal operator removes cargoes that are placed in high cost positions.

Two insertion operators are used in this algorithm: basic greedy insertion and regret- $k$  insertion. The regret- $k$  insertion first calculates the regret value, which is equal to the sum of the cost of inserting a cargo in  $K^{th}$  best position, for  $K = 2, \dots, k$ , minus the cost of inserting the cargo in its best position. In this paper,  $k$  is drawn randomly between 2 to 4. At each iteration, the regret- $k$  operator inserts the cargo with the highest regret value in its minimum cost position. This process continues until all cargoes have been inserted, with ties being broken by selecting the lowest cost insertion. In the basic greedy operator insertions are made on the basis of having the lowest cost only.

We use an adaptive weight adjustment in line 16 of Algorithm 2 to define and adjust the weight of each removal and insertion operators by using statistics from earlier iterations. To this end, the entire search is divided into segments of 100 iterations. We then calculate a score for each of the operators during the search in the current segment, and at the end of the segment new weights are calculated based on their performances. After weighting the operators, we select one of them using a roulette wheel mechanism. The removal and insertion operators are selected independently. We have implemented a simulated annealing acceptance criterion. The ALNS heuristic stops after 1000 iterations. For additional details we refer the readers to Hemmati et al. (2014).

### 3.6. New cargo quantity generator

New cargo quantities for the cargoes belonging to a homogeneous cargo set are generated in line 14 of Algorithm 1. The total quantity transported between the origin and destination ports in a homogeneous cargo set is held constant. The sum of the cargo

quantities in the homogeneous cargo set and the limits on the quantity unloaded or loaded in ports are considered when alternative quantities are generated for an individual cargo. To ensure that a constant total quantity is transported, new cargo quantities are generated for the first  $m - 1$  cargoes in a set, while the remaining quantity is assigned to the last cargo. If the quantity of the last cargo violates the minimum or maximum quantity limit, the quantities assigned to the  $m - 1$  first cargoes are infeasible and the combination is discarded.

The inputs of this algorithm are the current solution and the onboard quantities during the planning horizon. The algorithm also needs the capacity of the ships and the minimum and maximum limits on the quantity unloaded or loaded in ports. An example to illustrate the criteria used to calculate new cargo quantities now follows. Assume that  $C_g^{H_p^+}$  and  $C_g^{H_p^-}$  represent the loading and unloading visits of the cargo  $g$  in the homogeneous cargo set and  $C_g^{O^+}$  and  $C_g^{O^-}$  represent the loading and unloading visits of the ordinary cargo which does not belong to any homogeneous cargo set. Table 1 presents a feasible solution to an instance with two products, four ships with given capacities, two homogeneous cargo sets (one includes two cargoes and the other one includes three), and four ordinary cargoes. The first two ordinary cargoes and the first homogeneous cargo set handle the first product while the remaining cargoes handle the second product. The four ordinary cargoes have the following sizes: 620, 184, 860, and 210.

The solution presented contains the visit sequence, the unloaded or loaded quantities at each visit and the onboard quantity after each visit. For example, ship number 3 has a capacity of 1200 units. It starts its route by loading 184 units of the ordinary cargo number 2 and then loads the ordinary cargo number 1 with the quantity of 620. Thus, the onboard quantity is equal to 804. The ship continues its route to unload them: first cargo number 1 and then cargo number 2, after which there are no more cargoes onboard. These four visits constitute a *segment*, i.e., a list of successive cargoes in a route from the time at which the ship is empty until the time at which it becomes empty again. Thus, a route consists of one or more segments. The ship then loads the fourth ordinary cargo and the second cargo in the second homogeneous cargo set and unloads them respectively. These four visits also set up another segment in this route.

In the following we describe six criteria for generating alternative cargo quantities for homogeneous cargo sets, and we illustrate their use for the example of Table 1. In the example, among five cargoes related to homogeneous cargo sets, one is selected to explain all six criteria. The selected cargo, which has loading and unloading visits  $C_2^{H_2^+}$  and  $C_2^{H_2^-}$ , is the second cargo of the second homogeneous cargo set and is serviced by ship 3.

**Criterion 1: Minimum and maximum possible quantities.** The idea behind this criterion is that the extreme points sometimes provide an opportunity to handle more cargoes in a route and thus exploit the ship capacity efficiently. This mostly happens when the capacities of two ships are different. The minimum and maximum amounts ( $Q_{jk}^-$ ,  $Q_{jk}^+$ ) are input parameters. For the given example they are (250, 1000).

**Criterion 2: Equal quantities.** Sometimes equal quantities for all cargoes in a homogeneous cargo set yield the best route, such as when the capacities of two ships are the same or when extreme points lead to inefficient routes. For the second homogeneous cargo set in the example, the equal quantity is 690.

**Criterion 3: Ship capacities.** The idea is to use the full capacity of a ship. In the example, the only ship capacity between 250 and 1000 is 900, which is for ships 1 and 4.

**Criterion 4: Ship capacity subtracted by the maximum load of segments in the same ship.** Segments are defined by the onboard quantities of a given route. The idea of this criterion is to use the ship

**Algorithm 3** *k*-means clustering algorithm.

- 1: Choose *k* random points to set as cluster centers
- 2: Assign each object to the closest cluster center
- 3: When all objects have been assigned, recalculate the positions of the centers
- 4: Go back to Step 2 unless the centers have not changed

**Table 1**

Feasible solution to an instance with two products, four ships, two homogeneous cargo set and four ordinary cargoes.

V = 1	Capacity: 900	Route 1:	$C_1^{H_2^+}$	$C_1^{H_2^-}$	$C_3^{H_1^+}$	$C_3^{H_1^-}$			
		(Un)loading size:	690	–690	690	–690			
		Onboard:	690	0	690	0			
V = 2	Capacity: 1200	Route 2:	$C_2^{H_1^+}$	$C_2^{H_1^-}$	$C_3^{O^+}$	$C_3^{O^-}$			
		(Un)loading size:	690	–690	860	–860			
		Onboard:	690	0	860	0			
V = 3	Capacity: 1200	Route 3:	$C_2^{O^+}$	$C_1^{O^+}$	$C_1^{O^-}$	$C_2^{O^-}$	$C_2^{H_2^+}$	$C_4^{O^-}$	$C_2^{H_2^-}$
		(Un)loading size:	184	620	–620	–184	210	690	–210
		Onboard:	184	804	184	0	210	900	690
V = 4	Capacity: 900	Route 4:	$C_1^{H_1^+}$	$C_1^{H_1^-}$					
		(Un)loading size:	690	–690					
		Onboard:	690	0					

capacity in a segment as much as possible. For cargoes that are not in the segment, a new cargo quantity is defined to allow using the full ship capacity in the segment. This criterion is implemented for all ships and segments. In the example, this criterion can be applied to the first segment of Route 3 since there are no cargoes belonging to any of the homogeneous cargo sets in this segment. The ship capacity, 1200, minus the maximum onboard quantity of the first segment 804 is equal to 396.

*Criterion 5: Ship capacity, plus the initial cargo quantity, minus the maximum value of the segment.* This criterion is used for the second segment in the example which includes the given cargo. The idea is to increase the initial quantity of the cargo and use the whole capacity of the ship. In addition to exploiting the whole ship capacity in a segment, this criterion provides the opportunity of having cargoes (the remaining cargoes in the homogeneous cargo set) of smaller size, which can fit into other routes. The ship capacity, 1200, plus initial quantity 690, minus the maximum onboard quantity of the second segment, 900, is equal to 990.

*Criterion 6: Ship capacities minus the quantity of one ordinary cargo.* The idea behind this criterion is to use the full capacity of the ship by matching a cargo related to a homogeneous cargo set with an ordinary cargo. This criterion considers the quantity of ordinary cargoes (here 620, 184, 860 and 210), and subtracts these amounts from all ship capacities and keep those between 250 and 1000. This gives potential cargo size of 280, 340, 580, 690, 716, and 990.

Thus, after applying all six criteria, the second cargo of the second homogeneous cargo set gets the following list of alternative quantities: 250, 280, 340, 396, 580, 690, 716, 900, 990, and 1000.

### 3.7. *k*-means clustering algorithm

All feasible combinations of new quantities (see Section 3.6) are generated at line 17 of Algorithm 1. Each combination represents a ship routing and scheduling problem which is solved by the ALNS heuristic. Since the number of these combinations is enormous, the ALNS is not applied to all possible cargo combinations. Instead, a systematic selection among the combinations is performed, as described here and in Section 3.8. The combinations are first clustered using *k*-means clustering based on a similarity measure. In the algorithm of Section 3.8 an adaptive selection method then handles the selection of clusters and the selection of a combinations from clusters.

The *k*-means heuristic (MacQueen, 1967) partitions objects into a predefined number *k* of clusters. The similarity measure in the

classic form of *k*-means clustering is the Euclidean norm. We first choose *k* cluster centers that are as far as possible from each other. Each combination is then assigned to its closest cluster center, based on the similarity measure. When this assignment process is over, a new center is calculated for each cluster using the points assigned to it. The assignment is then restarted using the new cluster centers. As a result of this loop, the *k* centers may change between iterations. When the centers do not move any more, a local minimum has been reached and the process terminates. This algorithm is described in Algorithm 3:

### 3.8. Cluster selection and cluster weight adjustment

This algorithm starts with equal weights for all clusters and then automatically adjusts the weights using statistics from earlier iterations in accordance with the idea of adaptive weight adjustment (Ropke & Pisinger, 2006). To this end, the search is divided into equal parts of 10 iterations each. Clusters are assigned new weights according to the scores they obtained in the current part of the search. The score of a cluster is increased based on the quality of the selected combination. When a combination from the selected cluster improves upon the best solution of that cluster, the weight of that specific cluster is increased. The cluster receives double score if the combination selected from that cluster improves the global best solution. Let  $\omega_{is}$  be the weight of cluster *i* in part *s*. The new weights which are considered during part *s* + 1 are calculated as

$$\omega_{i(s+1)} = \omega_{is} \cdot (1 - r) + r \cdot \frac{\pi_i}{\theta_i}, \quad (56)$$

where  $\pi_i$  is the score of cluster *i*,  $\theta_i$  is the number of times cluster *i* has been selected, and *r* is a predefined parameter used to balance the earlier weights and the new normalized scores. The values of  $\pi_i$  and  $\theta_i$  are reset between parts. After assigning weights to the clusters, we have *k* clusters with weights  $\omega_i$ ,  $i \in \{1, \dots, k\}$ . We then select cluster *j* with probability  $\omega_j / \sum_{i=1}^k \omega_i$  by using a roulette wheel mechanism. After picking a cluster and a combination from the selected cluster, as mentioned in Algorithm 1, we need to calculate the time windows for the new selected combination by the model of Section 3.4.

## 4. Computational results

Two instance sets were defined to evaluate the proposed HCGR heuristic. The first set aims at providing an overview of the performance of the HCGR heuristic compared with exact solutions. To



**Table 2**  
Comparison of exact method and the HCGR heuristic on first test instances.

$\bar{T}$	No. of ports	No. of ships	No. of products	No. of inventories	Density (percent)	Optimal value	Seconds	HCGR average seconds	HCGR average gap percent
720	4	2	1	4	100	327,037	26	0.64	0
720	4	2	1	4	100	367,408	23	0.63	0
720	4	2	1	4	100	396,685	29	0.62	0
720	4	2	1	4	100	412,060	36	0.58	0
720	4	2	1	4	100	328,049	128	0.62	0
720	4	2	1	4	100	210,031	19	0.60	0
720	8	4	2	8	50	694,445	49	2.14	0
720	8	4	2	8	50	723,722	55	2.14	0
720	8	4	2	8	50	739,097	62	2.11	0
720	8	4	2	8	50	655,086	154	2.15	0
720	8	4	2	8	50	695,457	151	2.05	0
720	12	6	3	12	33	949,128	81	3.51	0
720	12	6	3	12	33	865,117	173	3.40	0
720	12	6	3	12	33	989,499	78	3.44	0
720	12	6	3	12	33	905,488	170	3.51	0
720	12	6	3	12	33	1,018,776	84	3.52	0
720	16	8	4	16	25	1,232,525	196	6.06	0
720	16	8	4	16	25	1,261,802	202	6.10	0
720	16	8	4	16	25	1,277,177	209	6.09	0
720	16	8	4	16	25	1,302,173	199	6.13	0
720	16	8	4	16	25	1,317,548	206	6.03	0
720	20	10	5	20	20	1,831,239	242	9.76	0
720	20	10	5	20	20	1,713,221	134	9.79	0.02
720	20	10	5	20	20	1,629,210	225	9.73	0
720	20	10	5	20	20	1,644,585	232	10.11	0
720	20	10	5	20	20	1,714,233	235	9.76	0

**Table 3**  
Performance of the HCGR heuristic on second test instances. All instances have 16 ports and 10 ships.

#	$\bar{T}$	No. of products	No. of inventories	Density (percent)	Best found	Average gap (percent)	Average seconds	Cargoes generated	No. of combinations
1	720	2	16	50	197,919	0	10	14	6
2	720	2	24	75	245,719	0.82	47	22	36
3	720	2	32	100	279,574	0	106	28	48
4	720	3	16	33	208,413	0.92	20	16	64
5	720	3	24	50	243,845	0.78	85	24	216
6	720	3	36	75	362,039	0	172	33	216
7	720	3	48	100	418,542	0.06	288	42	216
8	1080	2	16	50	307,975	6.06	55	23	13,824
9	1080	2	24	75	448,366	1.49	214	36	100,000
10	1080	2	32	100	574,165	9.67	390	46	100,000
11	1080	3	16	33	339,492	5.66	133	29	100,000
12	1080	3	24	50	436,105	1.32	262	41	100,000
13	1080	3	36	75	667,320	4.10	496	54	100,000
14	1080	3	48	100	869,273	6.84	886	70	100,000
15	1440	2	16	50	503,180	2.16	79	27	100,000
16	1440	2	24	75	797,267	7.75	278	44	100,000
17	1440	2	32	100	1,087,669	9.67	557	54	100,000
18	1440	3	16	33	545,973	2.98	223	37	100,000
19	1440	3	24	50	798,137	6.61	407	53	100,000
20	1440	3	36	75	1,269,740	3.41	825	66	100,000
21	1440	3	48	100	1,605,529	5.36	922	82	100,000

this end, we have generated small test instances that we were able to solve to optimality. CPLEX 12.4 was used to compute the optimal solutions to the arc-flow model of Section 2 as well as to the models of Sections 3.3 and 3.4. The first six instances of Table 2 were solved to optimality, but these six instances are very small. Thus, in order to evaluate the proposed algorithm for larger instances, we have combined small instances in such a way that the optimal value of the problem is the sum of the optimal solution values of the small instances. To achieve this, we have limited the compatibility between ships and ports. This structural information is not used directly by the HCGR heuristic. The second instance sets, with results presented in Tables 3 to 5, were generated to reflect the operation of short-sea shipping companies operating in Europe on the basis of realistic data, as shown in Fig. 1. This second set was

generated to analyze the performance of the algorithm on realistic size instances, and to evaluate the effect of attributes of instances on the solution quality and the running times of the HCGR.

As is characteristic of short-sea shipping, the loading and unloading times are large compared to the transportation times. In our instances, the ratio of loading and unloading times to the transportation times is set to 0.5. Moreover, the differences between the lower ( $S_{ik}$ ) and upper limits ( $\bar{S}_{ik}$ ) on the inventory levels are set approximately equal to the capacity of the smallest ship for consumption sites, and about twice as large for production sites. All inventories are half-full at the beginning of the planning horizon. There are more consumption sites than production sites, with about three consumption sites per production site. The test instances include seven different ship sizes, all in the handy size

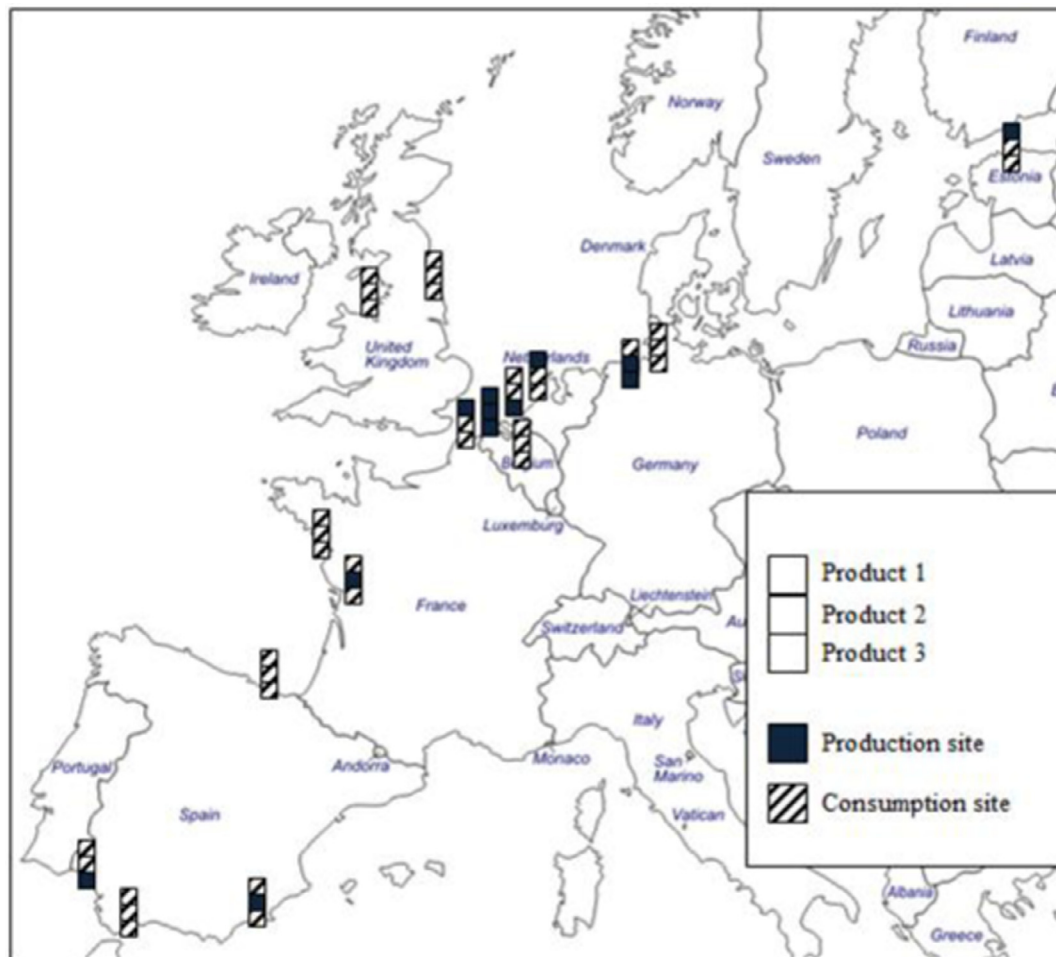


Fig. 1. Inventory sites for the instances with 16 ports and three products.

segment, ranging from 13,000 to 33,200 tons deadweight. Hemmati et al. (2014) defined the operating costs for these ships. Three different planning horizons of 720, 1080, and 1440 hours with three different numbers of ports, 16, 20, and 24, as well as different numbers of products are considered to analyze their effects on the performance of the algorithm. Both CPLEX and the HCGR heuristic were executed on a single core, using a 2.8 gigahertz CPU with 8 gigabyte RAM and a 64-bit operating system.

Table 2 provides results obtained from running the arc-flow model and the proposed HCGR heuristic on the first instance set, while Tables 3 to 5 provide the results obtained by executing the HCGR heuristic on the second instance set. The first four columns of Table 2 show the length of the planning horizon, the number of ports, the number of ships, and the number of products, respectively. The number of active inventories and the density are given in the next two columns. The density is the percentage of active inventories to the total number of potential inventories and is calculated as

$$\frac{(\text{Number of Inventories}) \cdot 100 \%}{(\text{Number of Ports}) \cdot (\text{Number of Products})}$$

Tables 2 to 5 are based on running the transportation model without any perturbations in the objective function coefficients. However, the best known solutions and the associated gaps were obtained after applying perturbations to the objective function coefficients of the transportation model. Table 2 provides the optimal

objective values for the first instance set. These are compared with the values obtained by the proposed HCGR heuristic. It is only fair to compare the running times of the exact method with the HCGR heuristic for the first six instances, because here no extra information is available for the exact method. For the remaining instances the running time is equal to the sum of the running times of a subset of first six instances. We have also applied the exact algorithm on the remaining instances in Table 2 without any extra information. However, after 3600 seconds, no feasible solution was found even when activating the feasibility setting, and the lower bounds were extremely weak. The last two columns in Table 2 are the average running times and the average gap to optimal values over five runs of the HCGR heuristic. The HCGR heuristic can find the same solution as the exact method for almost all instances of the first set, while its running time is significantly smaller than that of the exact method with extra information. However, we note that the instances in Table 2 have a particular structure which may favor the heuristic.

Tables 3 to 5 present the results of running the HCGR heuristic five times on each instance of the second set. The average gap in Tables 3 and 4 are based on the best known solution of the three cases presented in Table 4, plus the original instances in Table 3. The last two columns in Tables 3 and 5 are based on the last run and denote the number of generated cargoes through lines 2 to 9 of Algorithm 1 and the number of combinations generated at line 17 in Algorithm 1. An upper limit of 100,000 was considered for generating the combinations, and the Max Iteration is set to 100 for lines 19 to 30 in Algorithm 1.

**Table 4**

Performance of the HCGR heuristic for the second test instances when different numbers of ships are available for each instance.

#	T	V	Best found (percent deviation to solution in Table 3)			Average time (percent deviation to time in Table 3)			No. of ships used in best found solution			Average gap (percent)		
			Case with $ V  - 1$	Case with $ V $	Case with $ V  + 1$	Case with $ V  - 1$	Case with $ V $	Case with $ V  + 1$	Case with $ V  - 1$	Case with $ V $	Case with $ V  + 1$	Case with $ V  - 1$	Case with $ V $	Case with $ V  + 1$
1	720	1	–	0.00	0.00	–	–10.0	–10.0	–	1	1	–	0.0	0.0
2	720	1	–	0.00	0.00	–	–6.4	–14.9	–	1	1	–	0.0	0.0
3	720	1	–	0.00	0.00	–	5.7	–6.6	–	1	1	–	0.0	0.0
4	720	1	–	0.00	0.00	–	–10.0	–10.0	–	1	1	–	0.0	0.0
5	720	1	–	0.00	0.00	–	–7.1	–11.8	–	1	1	–	0.0	0.0
6	720	2	INF	0.00	0.00	59.9	–8.7	–9.3	NA	2	2	NA	0.0	0.0
7	720	2	INF	0.00	0.00	85.4	–11.8	–15.0	NA	2	2	NA	0.0	0.0
8	1080	2	7.86	–1.59	4.37	50.9	0.0	–9.1	1	2	2	9.6	0.0	6.1
9	1080	2	INF	0.92	–0.85	44.4	–16.4	–25.7	NA	2	2	NA	1.8	0.0
10	1080	3	0.70	3.48	10.03	–9.7	–24.1	–29.5	2	2	3	0.7	3.5	10.0
11	1080	2	1.04	0.00	0.36	43.6	–10.5	–17.3	1	1	1	1.0	0.0	0.4
12	1080	2	INF	–0.57	0.03	109.5	–1.9	–9.9	NA	2	2	NA	0.0	0.6
13	1080	3	–1.67	0.02	1.75	25.4	3.2	–4.0	2	3	3	0.0	1.7	3.5
14	1080	3	INF	–0.85	1.77	38.9	18.4	–12.3	NA	3	4	NA	0.0	2.6
15	1440	3	INF	0.60	0.47	–13.9	–12.7	–20.3	NA	3	3	NA	0.6	0.5
16	1440	5	6.91	1.71	–3.86	–16.2	–11.9	–23.7	4	4	4	11.2	5.8	0.0
17	1440	6	–1.41	–0.03	–0.29	–10.6	–9.0	–27.8	5	5	5	0.0	1.4	1.1
18	1440	3	3.73	0.43	5.37	0.5	–6.7	–19.7	2	2	3	3.7	0.4	5.4
19	1440	4	INF	–1.12	–3.81	5.2	1.4	–6.14	NA	4	4	NA	2.8	0.0
20	1440	7	0.16	1.10	3.63	3.2	–1.9	–7.27	6	6	7	0.2	1.1	3.6
21	1440	7	9.93	4.15	0.59	15.0	10.4	–2.49	6	7	7	9.9	4.2	0.6
Average			3.03	0.39	0.93	27.0	–5.2	–13.9				4.0	1.1	1.6

**Table 5**

Performance of the HCGR heuristic on second test instances when the number of inventories is kept constant and the number of available ships is 10 for all instances.

$\bar{T}$	No. of ports	No. of products	No. of inventories	Density (percent)	Best known	Average gap	Average time	Cargoes generated	No. of combinations
720	20	3	48	80	455,873	0.16	306	42	216
720	20	4	48	60	448,687	0.65	271	42	252
720	24	2	48	100	442,637	0.54	226	40	36
720	24	3	48	67	432,288	0.00	337	42	630
720	24	4	48	50	449,081	0.13	322	42	252
1080	20	3	48	80	990,654	4.61	741	69	100,000
1080	20	4	48	60	981,301	1.23	842	71	100,000
1440	20	3	48	80	1,733,675	4.66	891	81	100,000
1440	20	4	48	60	1,576,487	2.08	1179	83	100,000

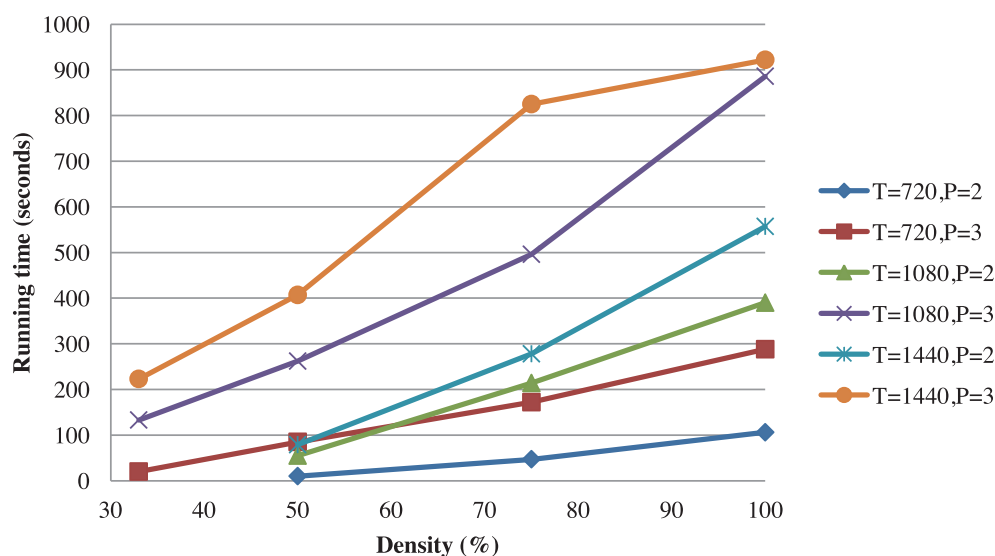
**Fig. 2.** Running time of the HCGR heuristic for different density settings.

Table 3 shows the performance of the HCGR heuristic on the second test instances when ten ships, of seven different sizes, are available for all instances. The total average gap of 3.6 percent illustrates that the proposed algorithm performs robustly, even for realistically sized instances. The running time is also acceptable for these instance sizes.

Fig. 2 illustrates how the parameters of the problem affect the running time of the proposed algorithm. It shows that for each combination of planning horizon length and number of products, increasing the density leads to larger running times. In addition, increasing the planning horizon and the number of products also increases the running time. To show the effect of the number of ships on the performance of the HCGR heuristic, the same instances as in Table 3 were solved again, but with only a subset of the vessels available. Three different numbers of available ships is considered in Table 4 where in the case with  $|V|$ , the ships are those appearing in the best solutions in Table 3. For the other two cases, either the smallest ship used in the best solution of Table 3 is removed ( $|V| - 1$ ) or one extra ship of the smallest size available is added ( $|V| + 1$ ). The number of ports in all instances in Table 4 is equal to 16 and the average gap is based on the best known solution of the three cases presented in Table 4 plus the best known solutions in Table 3. The columns for best found and average running time are shown using values relative to the corresponding values in Table 3.

The results of Tables 3 and 4 were analyzed to assess the effect of the number of inventories on the performance of the HCGR heuristic. Fig. 3 shows that increasing the number of inventories increases the HCGR heuristic running time but there is no significant relationship between the number of inventories and the gap.

Changing the number of ships does not affect the running time and the gap significantly. Fig. 3 suggests that the algorithm is not sensitive to the number of ships and can handle the fleet size and mix problem as well.

As Fig. 3 shows, the number of inventories apparently affects the complexity of the proposed algorithm. To analyze the effect of the rest of the parameters, the number of inventories was kept constant, and results for a subset of instances with 48 inventories are presented in Table 5, where the number of ports and products and consequently the density were allowed to vary.

Table 5 together with Tables 2–4 show that the number of ports does not affect the running time significantly, which means that the number of ports is not an adequate measure of instance size. It also shows that the planning horizon significantly affects the running time, as has already been illustrated in Fig. 2. We conclude from these results that the running time of the algorithm is mostly sensitive to the length of the planning horizon and to the number of active inventories. To illustrate this, Fig. 4 shows the relationship between the number of active inventories multiplied by the length of the planning horizon and the number of cargoes generated, as well as the relationship between the number of cargoes generated and the running time. The figure is based on the result of all 77 instances provided in Tables 2–5.

Finally, we have performed a sensitivity analysis to examine the effect of perturbing the objective function coefficients of the transportation model. To this end, we consider two sets of perturbation parameters. In the first set which is called “10 percent perturbation”,  $\varepsilon_{ij}$  is chosen randomly in  $[0, 0.1 C_{ij}]$  and  $\delta_{ij}$  is chosen to randomly in  $[0, (0.1 \cdot (C_{ik}^Q + C_{jk}^Q))/2]$ . The second set is called “20 percent perturbation” where  $\varepsilon_{ij}$  and  $\delta_{ij}$  are chosen randomly in



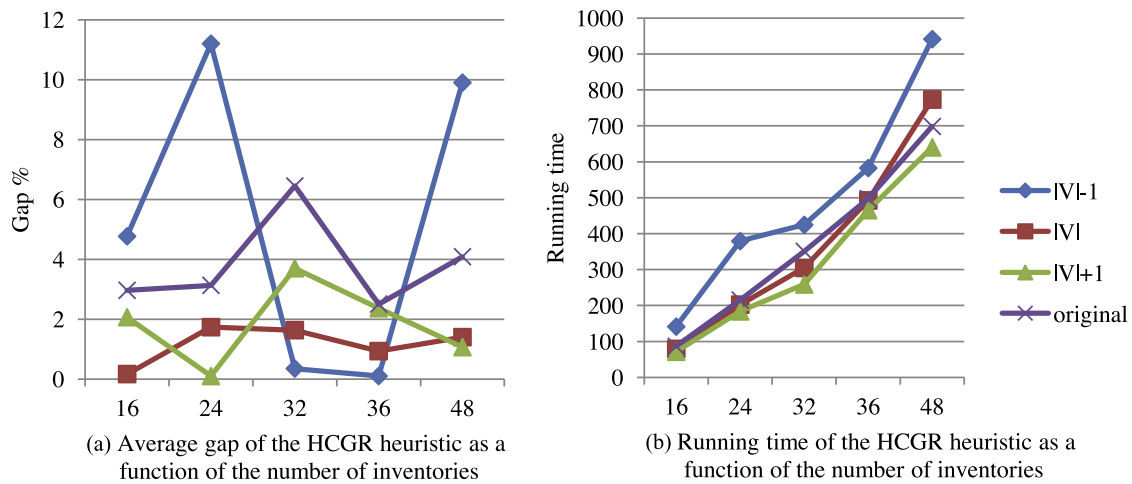


Fig. 3. Effect of the number of inventories.

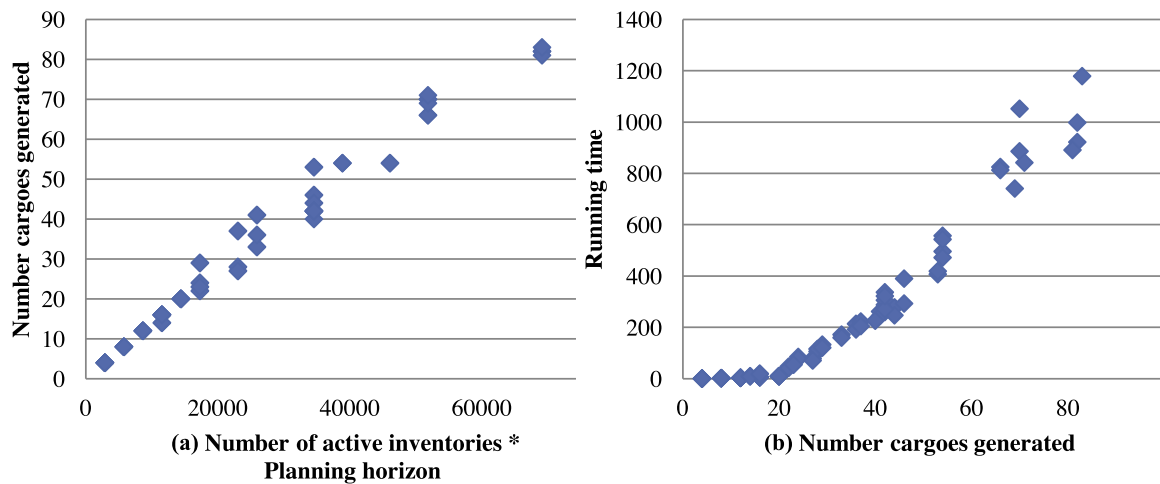


Fig. 4. Relationship between the number of active inventories multiple by planning horizon, the number of cargo generated and the running time of the HCGR heuristic.

Table 6

Performance of the HCGR heuristic by adding different perturbation parameters to the objective coefficients of the transportation model.

	No-perturbation	10 percent perturbation	20 percent perturbation
Average of best gap (percent) to the best know solutions	1.34	5.06	7.29
Average of average gap (percent) to the best know solutions	3.13	8.63	9.58
Number of contributions to the best known	33	7	5

$[0, 0.2 \cdot C_{ij}]$  and  $[0, (0.2 \cdot (C_{ik}^Q + C_{jk}^Q))/2]$  respectively. Table 6 shows the results of comparing these two sets with the no-perturbation objective coefficients of the transportation model. The results are based on 15 runs (five run for each of no -, 10 percent, and 20 percent perturbation) of all 51 instances in Tables 3 and Table 4 (case with  $|V|$ ), and Table 5. For each set of perturbation parameters the average of best gap to the best know solutions is reported together with the average of average gap to the best know solutions. The number of contributions of each set of perturbation parameters to the best known is also reported in Table 6.

Table 6 illustrates that perturbing the objective coefficients of the transportation model significantly downgrades the transportation model, and applying more perturbation results in worse solutions. Analyzing the solutions of 10 percent, and 20 percent perturbation with no-perturbation reveals that when we use perturbation, we obtain different cargoes in terms of origin, destination and quantity. The objective function of the transportation model in the no-perturbation mode is aligned with the objective function

of the main problem, thus it is not surprising to get worse results when we add perturbation although we get a chance of having different sets of cargoes each time.

## 5. Concluding remarks

We have modeled and solved a complex multi-product short sea inventory-routing problem faced by shipping companies. Some features of the problem, such as multiple products and a many-to-many distribution structure, make the problem very difficult to model and solve exactly. The instances that can be solved to optimality by CPLEX are rather small and not representative of real-life instances. Therefore, to be able to solve larger realistic instances, we have developed an effective and efficient iterative two-phase hybrid matheuristic called hybrid cargo generating and routing (HCGR). The proposed algorithm was compared with an exact solution method on small size instances. Our computational results show that the proposed HCGR heuristic can find an optimal

solution to almost all the small instances within seconds. Our results also show that the proposed heuristic is robust and can solve large and realistic instances within reasonable computing times. The performance of the algorithm was analyzed and it was observed that the running time of the algorithm is mostly sensitive to the length of the planning horizon and to the number of active inventories. We have also performed a sensitivity analysis to examine the effect of adding perturbations to the objective coefficients of the transportation model. It revealed that although perturbations provide the opportunity of generating different cargoes in the transportation model, the results are on average better when the objective functions of the transportation model and of the main problem are suitably aligned, even if this means that the cargoes generated are fixed throughout the execution of the heuristic.

## Acknowledgments

This research was carried out with financial support from the Research Council of Norway through the DOMinant II project, and from the Canadian Natural Sciences and Engineering Research Council under grant 39682-10. This support is gratefully acknowledged. Thanks are due to the referees for their valuable comments.

## References

- Agra, A., Christiansen, M., & Delgado, A. (2013). MIP formulations for a short sea fuel oil distribution problem. *Transportation Science*, 47(1), 108–124.
- Agra, A., Christiansen, M., Delgado, A., & Simonetti, L. (2014). Hybrid heuristics for a maritime short sea inventory routing problem. *European Journal of Operational Research*, 236(3), 924–935.
- Agra, A., Andersson, H., Christiansen, M., & Wolsey, L. (2013). A maritime inventory routing problem: Discrete time formulations and valid inequalities. *Networks*, 62(4), 297–314.
- Al-Khayyal, F., & Hwang, S.-J. (2007). Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. *European Journal of Operational Research*, 176(1), 106–130.
- Andersson, H. (2011). A maritime pulp distribution problem. *INFOR*, 49(2), 125–138.
- Andersson, H., Hoff, A., Christiansen, M., Hasle, G., & Løkketangen, A. (2010). Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9), 1515–1536.
- Bilgen, B., & Ozkaraahan, I. (2007). A mixed-integer linear programming model for bulk grain blending and shipping. *International Journal of Production Economics*, 107(2), 555–571.
- Bredström, D., Carlsson, D., & Rönnqvist, M. (2005). A hybrid algorithm for distribution problems. *IEEE Intelligent Systems*, 20(4), 19–25.
- Christiansen, M. (1999). Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1), 3–16.
- Christiansen, M., & Fagerholt, K. (2014). Ship routing and scheduling in industrial and tramp shipping. In P. Toth, & D. Vigo (Eds.), *Vehicle routing: problems, methods, and applications* (second ed.) (pp. 381–408). Philadelphia: MOS-SIAM series on optimization. Society for Industrial & Applied Mathematics, U.S.
- Christiansen, M., Fagerholt, K., Flatberg, T., Haugen, Ø., Kloster, O., & Lund, E. H. (2011). Maritime inventory routing with multiple products: A case study from the cement industry. *European Journal of Operational Research*, 208(1), 86–94.
- Christiansen, M., Fagerholt, K., Nygreen, B., & Ronen, D. (2013). Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3), 467–483.
- Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2014). Thirty years of inventory-routing. *Transportation Science*, 48(1), 1–19.
- Dauzère-Pères, S., Nordli, A., Olstad, A., Haugen, K., Koester, U., Myrstad, P. O., Teistklub, P., & Reistad, A. (2007). Onya Hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to European paper manufacturers. *Interfaces*, 37(1), 39–51.
- Engineer, F. G., Furman, K. C., Nemhauser, G. L., Savelsbergh, M. W. P., & Song, J.-H. (2012). A branch-price-and cut algorithm for single-product maritime inventory routing. *Operations Research*, 60(1), 106–122.
- Eurostat (2014). European Commission. [http://ec.europa.eu/eurostat/statistics-explained/index.php?title=Maritime\\_transport\\_statistics\\_-\\_short\\_sea\\_shipping\\_of\\_goods](http://ec.europa.eu/eurostat/statistics-explained/index.php?title=Maritime_transport_statistics_-_short_sea_shipping_of_goods) (April 2014)
- Fodstad, M., Uggen, K. T., Rømo, F., Lium, A.-G., Stremersch, G., & Hecq, S. (2010). LNGScheduler: A rich model for coordinating vessel routing, inventories and trade in the LNG supply chain. *Journal of Energy Markets*, 3(4), 31–64.
- Furman, K. C., Song, J.-H., Kocis, G. R., McDonald, M. M., & Warrick, P. H. (2011). Feedstock routing in the ExxonMobil downstream sector. *Interfaces*, 41(2), 149–163.
- Goel, V., Furman, K. C., Song, J.-H., & El-Bakry, A. S. (2012). Large neighborhood search for LNG inventory routing. *Journal of Heuristics*, 18(6), 821–848.
- Gronhaug, R., Christiansen, M., Desaulniers, G., & Desrosiers, J. (2010). A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science*, 44(3), 400–415.
- Gunnarsson, H., Rönnqvist, M., & Carlsson, D. (2006). A combined terminal location and ship routing problem. *Journal of the Operational Research Society*, 57(8), 923–938.
- Hemmati, A., Hvattum, L. M., Fagerholt, K., & Norstand, I. (2014). Benchmark suite for industrial and tramp ship routing and scheduling problems. *INFOR*, 52(1), 28–38.
- Hemmati, A., Stålhane, M., Hvattum, L. M., & Andersson, H. (2015). An effective heuristic for solving a combined cargo and inventory routing problem in tramp shipping. *Computers & Operations Research*, 64, 274–282.
- Hewitt, M., Nemhauser, G. L., Savelsbergh, M. W. P., & Song, J.-H. (2013). A branch-and-price guided search approach to maritime inventory routing. *Computers & Operations Research*, 40(5), 1410–1419.
- Li, J., Karimi, I. A., & Srinivasan, R. (2010). Efficient bulk maritime logistics for the supply and delivery of multiple chemicals. *Computers & Chemical Engineering*, 34(12), 2118–2128.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–297). University of California Press.
- Papageorgiou, D. J., Nemhauser, G. L., Sokol, J., Cheo, M.-S., & Keha, A. B. (2014). MIRPLib – A library of maritime inventory routing problem instances: Survey, core model, and benchmark results. *European Journal of Operational Research*, 234(2), 350–366.
- Persson, J. A., & Göthe-Lundgren, M. (2005). Shipment planning at oil refineries using column generation and valid inequalities. *European Journal of Operational Research*, 163(3), 631–652.
- Rakke, J. G., Stålhane, M., Moe, C. R., Andersson, H., Christiansen, M., Fagerholt, K., & Norstad, I. (2011). A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. *Transportation Research, Part C*, 19(5), 896–911.
- Rakke, J. G., Andersson, H., Christiansen, M., & Desaulniers, G. (2013). A new formulation based on customer delivery patterns for a maritime inventory routing problem. *Transportation Science*, 49(2), 384–401.
- Rocha, R., Grossmann, L. E., & de Aragão, M. V. P. (2013). Cascading knapsack inequalities: Reformulation of a crude oil distribution problem. *Annals of Operations Research*, 203(1), 1–18.
- Ronen, D. (2002). Marine inventory routing: Shipments planning. *Journal of the Operational Research Society*, 53(1), 108–114.
- Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4), 455–472.
- Shen, Q., Chu, F., & Chen, H. (2011). A Lagrangian relaxation approach for a multimode inventory routing problem with transshipment in crude oil transportation. *Computers & Chemical Engineering*, 35(10), 2113–2123.
- Sherali, H. D., & Al-Yakoob, S. M. (2006). Determining an optimal fleet mix and schedules: Part I – Single source and destination. In J. K. Karlof (Ed.), *Integer programming theory and practice* (pp. 137–166). Boca Raton, Florida: CRC Press.
- Sherali, H. D., & Al-Yakoob, S. M. (2006). Determining an optimal fleet mix and schedules: Part II – Multiple sources and destinations, and the option of leasing transshipment depots. In J. K. Karlof (Ed.), *Integer programming theory and practice* (pp. 167–194). Boca Raton, Florida: CRC Press.
- Siswanto, N., Essam, D., & Sarker, R. (2011). Solving the ship inventory routing and scheduling problem with undedicated compartments. *Computers & Industrial Engineering*, 61(2), 289–299.
- Song, J.-H., & Furman, K. C. (2013). A maritime inventory routing problem: Practical approach. *Computers & Operations Research*, 40(3), 657–665.
- Stålhane, M., Rakke, J. G., Moe, C. R., Andersson, H., Christiansen, M., & Fagerholt, K. (2012). A constructive and improvement heuristic for a liquefied natural gas inventory routing problem. *Computers & Industrial Engineering*, 62(1), 245–255.
- Uggen, K., Fodstad, M., & Nørstebo, V. (2013). Using and extending fix-and-relax to solve maritime inventory routing problems. *TOP*, 21(2), 355–377.