A deteriorating inventory routing problem for an inland liquefied natural gas distribution network

Ghiami, Yousef; Demir, Emrah; Van Woensel, Tom; Christiansen, Marielle; Laporte, Gilbert

published in
Transportation Research. Part B, Methodological
2019

DOI (link to publisher)
10.1016/j.trb.2019.05.014

document version
Publisher's PDF, also known as Version of record

Transportation Research Part B 126 (2019) 45–67

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

A deteriorating inventory routing problem for an inland liquefied natural gas distribution network

Yousef Ghiami[a,*], Emrah Demir[b], Tom Van Woensel[c], Marielle Christiansen[d], Gilbert Laporte[e]

[a] Department of Supply Chain Analytics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
[b] Panalpina Centre for Manufacturing and Logistics Research, Cardiff Business School, Cardiff University, Cardiff, United Kingdom
[c] School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
[d] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway
[e] Canada Research Chair in Distribution Management and Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation, HEC Montréal, Montréal, Canada

ARTICLE INFO

Article history:
Received 17 July 2018
Revised 20 May 2019
Accepted 23 May 2019
Available online 6 June 2019

Keywords:
Liquefied natural gas
Deteriorating item
Inventory routing problem
Matheuristic

ABSTRACT

Liquefied Natural Gas (LNG) is natural gas that is converted into its liquid state for logistical benefits. It is also becoming a more viable alternative energy source due to its price competitiveness and environmental friendliness. We study an inventory routing problem for inland distribution of LNG from a storage facility to several filling stations. Here, a transport planner is responsible for the inventory management at the storage facility and filling stations, as well as for the routing and scheduling of a heterogeneous fleet of vehicles. LNG evaporates at a constant rate over time at the storage facility and at the filling stations, and this characteristic relates to the inventory management problems with deterioration described in the literature. Therefore, the problem under study is denoted an LNG Deteriorating Inventory Routing Problem (LNG-DIRP). As a solution method, we propose a matheuristic that combines a mixed integer programming formulation and an adaptive large neighborhood search algorithm. Results of extensive computational experiments confirm the efficiency of the proposed solution method and provide managerial insights to promote LNG as an alternative clean energy solution for the future of transportation. Furthermore, we examine several replenishment policies that are of interest to practitioners and regulatory bodies.

1. Introduction

In the last decade, the ever increasing concern about global warming has been the main driver for regulators to tighten the regulations on greenhouse gases (GHGs) emissions. This changing environment has been a motivation for various industries to look for cleaner sources of energy. Liquefied natural gas (LNG) seems to be an appropriate alternative fuel not only for low levels of emissions, but also due to its low price compared with conventional fuels. A study of the global market of LNG predicts an annual increase of 4% in the demand until 2035 (Shell, 2018).

* Corresponding author.
E-mail addresses: y.ghiami@vu.nl (Y. Ghiami), demire@cardiff.ac.uk (E. Demir), t.v.woensel@tue.nl (T. Van Woensel), marielle.christiansen@iot.ntnu.no (M. Christiansen), gilbert.laporte@cirrelt.ca (G. Laporte).

https://doi.org/10.1016/j.trb.2019.05.014
0191-2615/

Driven by the growing LNG market, we have witnessed an increasing research effort on LNG-related optimization problems. The LNG literature has mainly focused on maritime transportation problems in which LNG is delivered from suppliers to ports close to markets (Christiansen et al., 2013). In order to make the storage and transportation of the natural gas more efficient and economical in such distribution networks, the temperature of the gas is decreased to approximately −162 degrees Celsius, at which point it transforms into its liquid state. Its volume is subsequently reduced by a factor of 610. Since LNG is kept at a boiling state throughout the distribution network, a percentage of the on-hand quantity constantly evaporates, which is called 'boil-off' gas. In the literature of LNG maritime transportation, researchers often assume that the amount of gas evaporated each day is equal to 0.1%–0.15% of the capacity of the LNG tank, see, e.g., Gerdsmeyer and Isalski (2005), Grønhaug et al. (2010) and Rakke et al. (2011). Examples of LNG-related applications can be found in Fodstad et al. (2010), Grønhaug et al. (2010), Rakke et al. (2011), Stålhane et al. (2012), Halvorsen-Weare et al. (2013), Uggen et al. (2013), Papageorgiou et al. (2014), Rakke et al. (2015), Shao et al. (2015), Al-Haidous et al. (2016), Mutlu et al. (2016), Andersson et al. (2016) and Andersson et al. (2017).

As previously discussed, LNG is kept at a temperature close to its boiling temperature which results in a constant evaporation of LNG. Due to the evaporation process of LNG, this commodity can be categorized as a *deteriorating item*, i.e. a product of which a percentage of on-hand inventory is constantly lost due, for instance, to decay, evaporation, or spoilage. It should be noted that in the distribution network considered in this study, the evaporated LNG, i.e. natural gas (NG), is considered as waste since it cannot be sold to the same market. However, NG could be transformed into, e.g., compressed natural gas (CNG) and sold through other channels. For an overview of deteriorating inventory we refer to Bakker et al. (2012) and Janssen et al. (2016). Except for LNG-related applications, there is limited literature on deteriorating item inventory-routing problems (IRPs), see Ghiami et al. (2015).

The inventory-routing problem (IRP) is defined at the interface of inventory management and the vehicle routing problem (VRP). It combines inventory and routing decisions. IRPs have gained increasing attention in recent years. For overviews, see Bertazzi et al. (2008) and Coelho et al. (2014). Recent studies on IRP can be found in Chitsaz et al. (2016), Hemmati et al. (2016), Roldán et al. (2016), Zhang and Unnikrishnan (2016), Agra et al. (2018), Soysal et al. (2018) and Rohmer et al. (2019).

As mentioned, the focus of the LNG literature is mainly on ship routing and scheduling, and little attention has been paid to inland LNG distribution (see, e.g., Christiansen et al., 2013 and Coelho et al., 2014). In this research, we focus on the inland distribution network of LNG that receives the inflow of LNG through deep sea shipping. In these operations, the evaporation of the liquid takes place at all facilities, and due to the scale of the operations and technical limitations, it is higher than in the maritime transportation. We denote this deteriorating inventory routing problem as LNG-DIRP, for which we present an advanced solution methodology. The LNG-DIRP consists of finding a set of routes for a set of heterogeneous vehicles, including barges and trucks, to deliver LNG quantities from a storage facility to a set of geographically scattered filling stations over a multi-period planning horizon.

This paper develops an arc-flow mathematical formulation for the LNG-DIRP. Since the problem at hand is NP-hard, we propose a matheuristic to solve it. This matheuristic combines both a simplified version of the mathematical model and the well-known adaptive large neighborhood search (ALNS) metaheuristic, introduced by Ropke and Pisinger (2006). The ALNS framework was initially put forward to solve variants of the VRP and has since been successfully implemented in many optimization problems (see, e.g., Demir et al., 2012; Ghilas et al., 2016; Franceschetti et al., 2017; SteadieSeifi et al., 2017).

To gain further practical insights, we study the LNG inland distribution network problem inspired by a real-life case study in the Netherlands. In this application, LNG is delivered from a storage facility to a group of filling stations geographically scattered all over the country, which fulfill the demands of end customers. Currently, this distribution system is at its very early stage of development. Hence, the existing operational data such as cost parameters or vehicle capacities could vary from one LNG transporter to another.

The contribution of this paper is threefold: (*i*) A mixed integer programming formulation of an LNG inventory routing problem, where evaporation of LNG is considered at all facilities; (*ii*) a new matheuristic that combines a mixed integer programming formulation and an adaptive large neighborhood for the LNG-DIRP; here, a new type of operators called break-move operators, dealing with the quantities in the IRP, is developed; (*iii*) an analysis on several replenishment policies.

The remainder of this paper is structured as follows. Section 2 introduces the LNG-DIRP. Section 3 presents a mixed-integer linear programming formulation for the problem. Section 4 describes the proposed matheuristic. Section 5 presents the extensive computational experiments. This is followed by conclusions and managerial insights in Section 6.

2. Problem description

We now describe the LNG distribution network starting from a storage facility located at a sea port and connecting it all the way to geographically dispersed filling stations. The supply of LNG to the storage facility is made through deep-sea shipping from upstream suppliers. The LNG received at the storage facility is used to fulfill the demand of the filling stations. The LNG holding cost is significant since both the storage facility and the filling stations must keep the temperature at a very low level to maintain the LNG in its liquid state.

The filling stations have a deterministic demand and a limited capacity. Their inventories are replenished from a storage facility which is the only upstream supplier in the network. Shortages are not allowed at the storage facility nor at the filling stations. In this distribution network, there are two types of vehicles, namely trucks and barges, which transport the

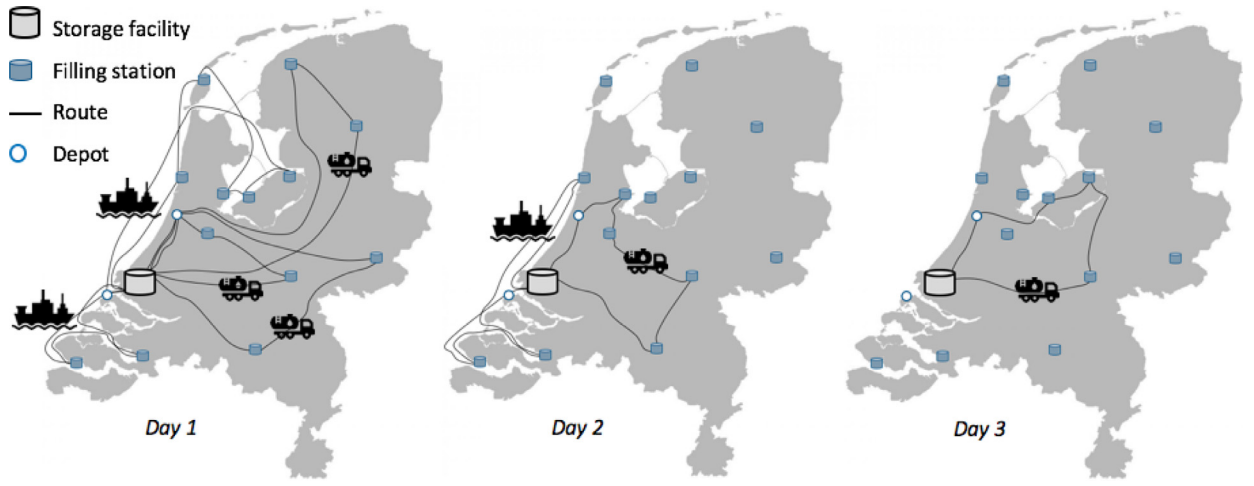

Fig. 1. Example of a three-day plan for the LNG inland distribution network in the Netherlands.

LNG between the storage facility and the filling stations. The trucks offer a high level of flexibility and accessibility with relatively high variable costs and a limited capacity, whereas barges can carry much larger quantities with a limited access and lower variable costs. In this problem, we assume that each type of carrier owns a limited number of homogeneous vehicles, and each vehicle type has a designated depot. Therefore, each route starts from one of the two depots, depending on which vehicle has been assigned to the route, and ends at the same depot when the vehicle has first visited the storage facility and then a set of filling stations.

The high operational costs of LNG are not confined to the maritime side of the supply chain and are present throughout the entire distribution network. Indeed, loading the LNG on vehicles at the storage facility and unloading it at the filling stations involve high fixed costs due to the technical complexity of the operations. Fig. 1 depicts an example of an operational plan for the LNG inland distribution network in the Netherlands with a planning horizon of three days. We assume that if an operation (loading, unloading, and transportation operations at the storage facility and the filling stations) is planned to take place on a day, it is executed at the start of that day. Since all locations have some storage capacity, they may receive quantities that are sufficient to cover the demand of more than one day, and if so, they carry an inventory from one day to the next.

Our aim is to maximize the total profit of the distribution network by determining optimal inventory and routing policies, considering both tactical and operational level challenges. The revenues of this integrated system are generated at the filling stations' level where the demand of the final customers is met. Since shortages are not allowed and the demand is deterministic, the profit maximization problem is equivalent to a cost minimization problem (Ghiami and Beullens, 2016). The total cost function of the problem consists of fixed purchasing costs at the filling stations and at the storage facility, variable purchasing costs at the storage facility, holding costs at the filling stations and the storage facility, routing costs, and vehicle fixed costs. The variable purchasing costs paid by the filling stations, are received by the storage facility as revenues. These transfer prices have no effect on total cost of the system and are therefore discarded from the objective function.

3. Mathematical formulation of the LNG-DIRP

We now model the LNG-DIRP by means of an arc-flow formulation. The LNG-DIRP is defined on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A}^m)$, where $\mathcal{N}$ is the set of all nodes in the network including nodes with storage capacity ($\mathcal{N}^{'}$) and depots ($\mathcal{N}^{''}$), $\mathcal{N} = \mathcal{N}^{'} \cup \mathcal{N}^{''} = \{0, \ldots, n+2\}$. In this set, node 0 represents the storage facility (pick-up point) and nodes 1 to n are the filling stations (delivery locations). We denote the set of the delivery locations by $\mathcal{N}^L = \{1, \ldots, n\}$ ($\mathcal{N}^{'} = \{0\} \cup \mathcal{N}^L$). In terms of transportation mode, vehicles are divided into two groups, road (1) and sea (2), which are presented by $\mathcal{M} = \{1, 2\}$. More specifically we use a set of r homogeneous trucks and w homogeneous barges, $\mathcal{V} = \{1, \ldots, r, r+1, \ldots, r+w\}$. These vehicles start and end their trip at a depot. Nodes $n+1$ and $n+2$ represent the depots for trucks and barges, respectively. Since the distance between a pair of nodes depends on the vehicle type that performs the trip, we define an arc set in this network that depends on the vehicle chosen for the route. The arc set for this graph is defined as $\mathcal{A}^m = \{(i, j)^m : i, j \in \mathcal{N}, i \neq j, m \in \mathcal{M}\}$, where $(i, j)^m$ represents the arc between nodes i and j using transportation mode m.

While stored in these facilities, θ percent of the LNG evaporates. However, we assume that the LNG evaporation during the transportation operations can be handled outside the optimization. More details will be given in Section 3.1. The set of time periods in the planning horizon is defined as $\mathcal{T} = \{1, \ldots, H\}$, where H is the latest planning day. In each time period a vehicle assigned to a route starts its trip from its depot after which it visits the storage facility. A predetermined quantity

is then loaded on the vehicle. Afterwards, the vehicle visits a set of filling stations and unloads the demanded quantities of LNG. The vehicle finally returns to its depot. In each time period, a vehicle can be used at most once. We define binary variables w_{ivt} equal to one if and only if vehicle v visits node i ($i \in \mathcal{N}'$) in period t. In period t vehicle v loads or unloads a quantity of q_{ivt}. We define binary variables x_{ijvt} ($(i,j)^m \in \mathcal{A}^m$) equal to one if and only if vehicle v visits node j immediately after node i in period t.

Vehicle v ($v \in \mathcal{V}$) has a capacity of $\bar{V}_v$ and a variable cost of C^T_{ijv} to operate between nodes i and j ($(i,j)^m \in \mathcal{A}^m$). If, due to infrastructural limitations, vehicle v cannot traverse arc (i,j), a suitably large value is assigned to C^T_{ijv}. We note that technical complexity of loading and unloading LNG results in a fixed cost of C^{FV}_v when vehicle v visits a node. In each trip, due to practical limitations, vehicle v can visit a maximum number of filling stations, $\bar{N}^L_v$.

We assume an inventory capacity of $\bar{S}_i$ and also a minimum inventory level of $\underline{S}_i$ at $i \in \mathcal{N}'$. Moreover, node $i \in \mathcal{N}'$ has an initial inventory S_i at the start of the planning horizon. The inventory level at node i at the end of period t is denoted by s_{it}. The binary variable z_t is equal to one if and only if the storage facility replenishes its inventory in period t. The storage facility orders a quantity of y_t from its upstream supplier at the start of period t. We assume the lead times for replenishment to be zero for both the storage facility and the filling stations. A total amount of $\sum_{v\in\mathcal{V}} q_{0vt}$ is loaded on vehicle v in period t. After replenishing its own inventory and also sending quantities to a group of delivery locations, the inventory level at the storage facility depletes at the rate of θ during time period t until the inventory level reaches s_{0t} at the end of the time period.

At the beginning of time period t, delivery location $i \in \mathcal{N}^L$ replenishes its inventory by an amount of $\sum_{v\in\mathcal{V}} q_{ivt}$. Afterwards, the inventory level decreases since the filling station meets the constant demand rate of D_{it} and faces an evaporation rate of θ.

3.1. Boil-off gas rate estimation

The boil-off gas (BOG) calculation is a complex task, and the amount of BOG depends on the design and operating conditions of LNG tanks and ships (Dobrota et al., 2013). The handling of BOG varies in different papers due to the underlying real planning problem and the focus of the paper.

Most of the research reported in the literature related to LNG inventory routing problems are developed for deep-sea shipping problems with a long planning horizon, where tankers carry huge quantities of LNG to distant destinations. Thus, these voyages may take as long as 30 days see, e.g., Halvorsen-Weare and Fagerholt (2013) and Halvorsen-Weare et al. (2013). For economical reasons, in LNG shipping, normally LNG carriers depart the pick-up ports fully loaded. Over such long trips, the quantity of the boil-off gas is significant. In this subset of the literature, researchers indicate different bases for measuring the quantity of boil-off gas.

Some assume that the boil-off gas rate is based on the LNG tank capacity (or the initial LNG quantity since tankers are usually fully loaded when departing a pick-up port), see, e.g., Grønhaug et al. (2010), Rakke et al. (2011) and Andersson et al. (2016). A group of researchers that consider the boil-off gas to be a percentage of the tank capacity, differentiate between ballast (when a ship is almost empty and moving towards a pick-up port for loading LNG) and laden (when an LNG tanker is fully loaded) voyages by assigning lower rates for boil-off to ballast voyage, see, e.g., Fodstad et al. (2010) and Dobrota et al. (2013). Several researchers assume that the boil-off is a function of the LNG quantity that is in the tank, see, e.g., Cho et al. (2018). By assuming a fixed daily amount of BOG depending on the capacity of the tank, a conservative estimate can be calculated. Depending on the underlying real planning problem, different assumptions are taken. By including the BOG in the optimization, they make sure that enough quantity is left in the tankers to make their ballast voyage, without running out of LNG. Moreover, LNG carriers are normally able to use the boil-off gas as fuel, see, e.g., Grønhaug et al. (2010), Goel et al. (2012) and Christiansen et al. (2013)

In practice, calculating the exact boil-off gas rate is complicated and depends on several factors. Moreover, it is an evolving field and there are recent papers that consider different factors. Some technical papers elaborate further on the conventional boil-off gas rate (0.10–0.15%) and specify under what condition this range would be obtained, see, e.g., Moon et al. (2007). Yet calculating the boil-off gas based on the initial quantity (or the capacity of the tank) in large scale operations over long voyages may give a sufficient estimation. For small scale operations and short transportation, however, a better estimation may need to be obtained. Some technical reports/papers suggest to consider the quantity in the tank as the base for boil-off gas calculation, see, e.g., Jones (1974), Shin et al. (2008), Gedde (2014) and Kim et al. (2019).

In our research, we have a storage facility and filling stations with a higher evaporation rate that for the LNG vessels studied in the literature, so we consider the evaporation rate to be a percentage of the quantity of LNG in the tank at these storage tanks. For the trucks and barges, the boil-off rates are lower. Due to short distances, the trucks and barges are in a ballast situation with low boil-off rates in much of each time period. Therefore, we assume that we can neglect the boil-off volume for the trucks and the barges in the optimization. Instead, the boil-off quantity per day is precalculated for each truck and barge, and taken into account in the parameter setting and the values of the variables.

3.2. Inventory level at nodes

In order to compute the inventory level at delivery point i at the end of period t, we define the inventory level as a function $s^L_{it}(u)$ of time throughout the time period where u is the time variable and varies between 0 to 1, representing the

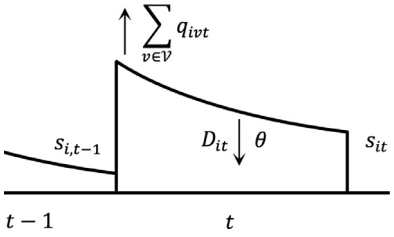

Fig. 2. The inventory level at filling station i during period t.

start and end of that period, respectively. This inventory level decreases during the period due to demand and deterioration (see Fig. 2). The following differential equation represents the changes in the inventory level during period t:

$$\frac{ds^L_{it}(u)}{du} = -\theta s^L_{it}(u) - D_{it} \quad i \in \mathcal{N}^L,\ t \in \mathcal{T},\ u \in [0,1]. \tag{1}$$

The inventory level at filling station i at the beginning of period t is the sum of the inventory level at the end of the previous period and the quantities delivered by the vehicles at the start of the current period, hence the following boundary condition is deduced:

$$s^L_{it}(u=0) = s_{i,t-1} + \sum_{v\in\mathcal{V}} q_{ivt} \quad i \in \mathcal{N}^L,\ t \in \mathcal{T}. \tag{2}$$

Solving the differential equation (1), using boundary condition (2), results in the inventory level of filling station i during time period t:

$$s^L_{it}(u) = -\frac{D_{it}}{\theta} + \left[s_{i,t-1} + \sum_{v\in\mathcal{V}} q_{ivt} + \frac{D_{it}}{\theta}\right] e^{-\theta u} \quad i \in \mathcal{N}^L,\ t \in \mathcal{T},\ u \in [0,1]. \tag{3}$$

The inventory levels of filling station i at the end of period 0 and t are therefore given by

$$s_{i0} = \left[S_i + \sum_{v\in\mathcal{V}} q_{iv0}\right] e^{-\theta} - \frac{D_{i0}}{\theta}\left(1 - e^{-\theta}\right) \quad i \in \mathcal{N}^L, \tag{4}$$

and

$$s_{it} = \left[s_{i,t-1} + \sum_{v\in\mathcal{V}} q_{ivt}\right] e^{-\theta} - \frac{D_{it}}{\theta}\left(1 - e^{-\theta}\right) \quad i \in \mathcal{N}^L,\ t \in \mathcal{T}, \tag{5}$$

respectively.

The inventory level at the storage facility changes in a different way since the demand from the customer side is fulfilled at the start of each period. This means that during each period the inventory level depletes only due to the deterioration. Fig. 3 and the following equations illustrate the change in the inventory level at the storage facility during time period t:

$$\frac{ds^P_{0t}(u)}{du} = -\theta s^P_{0t}(u) \quad t \in \mathcal{T},\ u \in [0,1]. \tag{6}$$

For time period t, the following boundary condition holds for the inventory level at the storage facility:

$$s^P_{0t}(u=0) = s_{0,t-1} + y_t - \sum_{v\in\mathcal{V}} q_{0vt} \quad t \in \mathcal{T}. \tag{7}$$

Using Eqs. (6) and (7), the inventory level of the storage facility during time period t is then computed as

$$s^P_{0t}(u) = \left[s_{0,t-1} + y_t - \sum_{v\in\mathcal{V}} q_{0vt}\right] e^{-\theta u} \quad t \in \mathcal{T},\ u \in [0,1]. \tag{8}$$

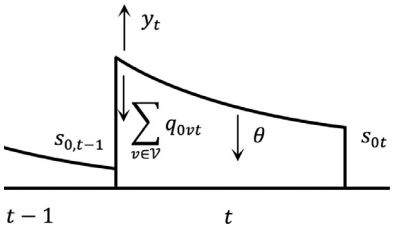

Fig. 3. The inventory level at the storage facility during period t.

The ending inventory levels at the pick-up point at time period 0 and t are then given by

$$s_{00} = \left[S_0 + y_0 - \sum_{v \in \mathcal{V}} q_{0v0} \right] e^{-\theta} \tag{9}$$

and

$$s_{0t} = \left[s_{0,t-1} + y_t - \sum_{v \in \mathcal{V}} q_{0vt} \right] e^{-\theta} \quad t \in \mathcal{T}, \tag{10}$$

respectively.

Using the inventory levels (4)–(5) and (9)–(10) obtained for the filling stations and the storage facility, in Section 3.3, we derive the objective function and the required constraints for the LNG-DIRP.

3.3. The objective function and constraints

The objective function of the LNG-DIRP consists of fixed and variable purchasing and holding costs at the storage facility, fixed purchasing and holding costs at the filling stations, and routing costs. The variable purchasing cost of the filling stations has no effect on the total cost of the system, hence it is discarded.

The storage facility receives a quantity y_t at the start of period t which results in the binary variable z_t getting a value of 1. This replenishment incurs a fixed cost of C_0^F and a variable cost of $C_{0t}^P y_t$ defined as

$$PC^P = \sum_{t \in \mathcal{T}} (C_0^F z_t + C_{0t}^P y_t). \tag{11}$$

The total fixed purchasing cost of the filling stations is given by

$$PC^L = \sum_{i \in \mathcal{N}^L} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}} C_i^F w_{ivt}. \tag{12}$$

Each vehicle visit to the facilities on the distribution network incurs a fixed cost which, together with the travel cost between nodes, yields the total routing cost of the LNG-DIRP:

$$RC = \sum_{(i,j) \in \mathcal{A}^m} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}} C_{ijv}^T x_{ijvt} + \sum_{i \in \mathcal{N}'} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}} C_v^{FV} w_{ivt}. \tag{13}$$

Keeping LNG at delivery point i costs a unit holding cost of C_i^H. Using the inventory level of the filling station during time period t, presented in (3), the following holding cost is calculated for the period:

$$HC_{it}^L = \int_0^1 C_i^H s_{it}^L(u) du = \frac{C_i^H \left[1 - e^{-\theta}\right]}{\theta} \left[s_{i,t-1} + \sum_{v \in \mathcal{V}} q_{ivt} + \frac{D_{it}}{\theta} \right] - \frac{C_i^H D_{it}}{\theta} \quad i \in \mathcal{N}^L,\ t \in \mathcal{T},\ t > 0. \tag{14}$$

The total holding cost of all the filling stations over the planning horizon is, therefore

$$HC^L = \sum_{i \in \mathcal{N}^L} \sum_{t \in \mathcal{T}} HC_{it}^L. \tag{15}$$

We calculate the holding cost at the storage facility in a similar way. First, the holding cost is calculated for period t:

$$HC_{0t}^P = \int_0^1 C_0^H s_{0t}^P(u) du = \frac{C_0^H \left[1 - e^{-\theta}\right]}{\theta} \left[s_{0,t-1} + y_t - \sum_{v \in \mathcal{V}} q_{0vt} \right] \quad t \in \mathcal{T},\ t > 0, \tag{16}$$

which results in a total holding cost of

$$HC^P = \sum_{t \in \mathcal{T}} HC_{0t}^P. \tag{17}$$

The holding cost at $t = 0$ for both the filling stations and the storage facility can be obtained using (14) and (16), respectively, by replacing $s_{i,t-1}$ with S_i ($i \in \mathcal{N}'$). Summing up the cost components of the total cost function of the LNG-DIRP, the objective function of the model is as follows:

$$\text{minimise } TC = PC^P + PC^L + RC + HC^L + HC^P. \tag{18}$$

subject to:

(4), (5), (9), (10), and

$$\sum_{v \in \mathcal{V}} q_{iv0} \leq \bar{S}_i - S_i \quad i \in \mathcal{N}^L, \tag{19}$$

$$s_{i,t-1} + \sum_{v \in \mathcal{V}} q_{ivt} \leq \bar{S}_i \quad i \in \mathcal{N}^L,\ t \in \mathcal{T}, \tag{20}$$

$$y_0 - \sum_{v\in\mathcal{V}} q_{0v0} \le \bar{S}_0 - S_0 \tag{21}$$

$$s_{i,t-1} + y_t - \sum_{v\in\mathcal{V}} q_{ivt} \le \bar{S}_i \quad i \in \mathcal{N}^P,\ t \in \mathcal{T}, \tag{22}$$

$$s_{it} \ge \underline{S}_i \quad i \in \mathcal{N}',\ t \in \mathcal{T}, \tag{23}$$

$$\sum_{i\in\mathcal{N}^L} w_{ivt} \le \bar{N}_v^L \quad v \in \mathcal{V},\ t \in \mathcal{T}, \tag{24}$$

$$\sum_{v\in\mathcal{V}} w_{ivt} \le 1 \quad i \in \mathcal{N}^L,\ t \in \mathcal{T}, \tag{25}$$

$$y_t \le (\bar{S}_0 - \underline{S}_0) z_t \quad t \in \mathcal{T}, \tag{26}$$

$$q_{0vt} = \sum_{i\in\mathcal{N}^L} q_{ivt} \quad v \in \mathcal{V},\ t \in \mathcal{T}, \tag{27}$$

$$q_{ivt} \le \min\{\bar{S}_i - \underline{S}_i, \bar{V}_v\} w_{ivt} \quad i \in \mathcal{N}',\ v \in \mathcal{V},\ t \in \mathcal{T}, \tag{28}$$

$$x_{i(v)0vt} = w_{0vt} \quad v \in \mathcal{V},\ t \in \mathcal{T},\ i(v) \in \mathcal{N}'', \tag{29}$$

$$w_{0vt} = \sum_{i\in\mathcal{N}^L} x_{ij(v)vt} \quad v \in \mathcal{V},\ t \in \mathcal{T},\ j(v) \in \mathcal{N}'', \tag{30}$$

$$\sum_{j\in\mathcal{N},(j,i)\in\mathcal{A}^m} x_{jivt} + \sum_{j\in\mathcal{N},(i,j)\in\mathcal{A}^m} x_{ijvt} = 2w_{ivt} \quad i \in \mathcal{N},\ v \in \mathcal{V},\ t \in \mathcal{T}, \tag{31}$$

$$\sum_{i\in\mathcal{N}^l} \sum_{j\in\mathcal{N}^l, i\ne j} x_{ijvt} \le \sum_{i\in\mathcal{N}^l} w_{ivt} - w_{kvt} \quad \mathcal{N}^l \subseteq \mathcal{N}^L,\ k \in \mathcal{N}^l,\ v \in \mathcal{V},\ t \in \mathcal{T}, \tag{32}$$

$$q_{ivt} \ge 0 \quad i \in \mathcal{N}',\ v \in \mathcal{V},\ t \in \mathcal{T}, \tag{33}$$

$$s_{it} \ge 0 \quad i \in \mathcal{N}',\ t \in \mathcal{T}, \tag{34}$$

$$y_t \ge 0 \quad t \in \mathcal{T}, \tag{35}$$

$$z_t \in \{0,1\} \quad t \in \mathcal{T}, \tag{36}$$

$$w_{ivt} \in \{0,1\} \quad i \in \mathcal{N}',\ v \in \mathcal{V},\ t \in \mathcal{T}, \tag{37}$$

$$x_{ijvt} \in \{0,1\} \quad (i,j) \in \mathcal{A}^m,\ v \in \mathcal{V},\ t \in \mathcal{T}. \tag{38}$$

Constraints (19)–(23) impose the allowed limits on the inventory level at all nodes. Constraints (24) control the number of filling stations that can be visited in each route. Constraints (25) limit the number of visits that a filling station can receive in each time period. Constraints (26) guarantee that the replenishment quantities received by the storage facility respects the storage capacity. On a route, the quantity loaded on a vehicle at the storage facility is equal to the sum of amounts delivered to the filling stations on that route. This is ensured by constraints (27). The quantity and routing decision variables are linked together by constraints (28). If a vehicle is assigned to a route, then the first arc of the route would be from the depot to the storage facility, and the last arc is between one of the filling stations and the depot. Constraints (29) and (30) construct that part of each route. Degree constraints and subtour eliminations constraints are enforced in (31) and (32), respectively. Lastly, constraints (33)–(38) ensure non-negativity and integrality conditions for the decision variables.

The LNG-DIRP is NP-hard since it is an extension of the classical IRP. Ghiami et al. (2015) have shown that only a simplified version of this problem can be solved to optimality for small-size instances. For this reason, we have developed a matheuristic to solve the problem and aim at obtaining good-quality solutions within reasonable solution times.

4. Solution methodology

This section provides the solution methodology proposed for the LNG-DIRP. Algorithm 1 provides the steps of the proposed methodology. The adaptive large neighborhood search was introduced by Ropke and Pisinger (2006) and has since been applied to several VRP applications. The main idea of ALNS, when applied to VRPs, is to start from an initial solution and in an iterative algorithm destroy the solution by removing some visits to customers and reinserting those visits into existing or new routes. The selection of the removal, break-move, and insertion operators is governed by a probabilistic mechanism which accounts for their past performance.

As shown in Algorithm 1, in the first step, an assignment problem is solved using the mathematical solver. The quantities obtained from the assignment problem are used to construct routes for a feasible initial solution for the ALNS. Removal

Algorithm 1: General framework of the matheuristic.

input : A set of removal operators R, a set of break-move operators B, a set of insertion operators I, cooling rate h, cooling factor P_{init}
output: X_{best}

Discard routing constraints and optimize the quantities by solving an assignment problem
Build feasible routes using the obtained quantities
Initialize probability P_d^t for destroy operator $d \in R$, probability P_b^t for break-move operator $b \in B$, and probability P_i^t for insertion operator $i \in I$
Let $Temp$ be the temperature and j be the counter initialized as $j \leftarrow 1$
Let $X_{current} \leftarrow X_{best} \leftarrow X_{init}$
repeat
  Select a removal operator $d^* \in R$ with probability P_d^t
  Let X_{new}^* be the solution obtained by applying operator d^* to $X_{current}$ and $\mathcal{L}$ be the list of removed visits
  Select a break-move operator $b^* \in B$ with probability P_b^t
  Let $\mathcal{L}'$ be the modified list of removed visits after applying operator b^* to $\mathcal{L}$
  Select an insertion operator $i^* \in I$ with probability P_i^t
  Let X_{new} be the new solution obtained by applying operator i^* to X_{new}^*
  if $c(X_{new}) < c(X_{current})$ **then**
    $X_{current} \leftarrow X_{new}$
  else
    Let $\zeta \leftarrow e^{-(c(X_{new})-c(X_{current}))/Temp}$
    Generate a random number $\lambda \in [0, 1]$
    if $\lambda < \zeta$ **then**
      $X_{current} \leftarrow X_{new}$
  if $c(X_{current}) < c(X_{best})$ **then**
    $X_{best} \leftarrow X_{new}$
  $Temp \leftarrow h.Temp$
  Update probabilities using the adaptive weight adjustment procedure
  $j \leftarrow j + 1$
until the maximum number of iterations is reached
Let $X_{current} \leftarrow X_{best}$
for all the existing routes **do**
  Create a tabu list
  repeat
    for all the possible pairs of filling stations on the route **do**
      Let X_{new} be the new solution obtained by swapping the position of the two filling stations in $X_{current}$
      if $c(X_{new}) < c(X_{current})$ **and** the move is not on the tabu list **then**
        $X_{current} \leftarrow X_{new}$
      Add the move that improved the solution to the tabu list
    if $c(X_{current}) < c(X_{best})$ **then**
      $X_{best} \leftarrow X_{current}$
  until the maximum number of iterations is reached

operators are then used to destroy the current solution by removing some visits from their current route. In the next step, break-move operators modify the removed quantities in terms of time and size by breaking them and partially or fully moving them forward or backward. Using the insertion operators, the modified removed quantities are reinserted into the destroyed solution, while satisfying the feasibility conditions. The algorithm replaces the current solution with the newly found solution if the new solution is better than the current solution. After a predetermined number of iterations, the algorithm adjusts the probabilities of the operators. If the stopping condition is not yet reached, the process is reiterated.

We apply simulated annealing to diversify the search within the feasible region. The variable X_{best} is the best solution found during the search, $X_{current}$ is the current solution obtained at the beginning of an iteration, and X_{new} is a temporary solution found at the end of an iteration, which can be discarded or become the current solution. The cost of solution X is denoted by $c(X)$. A candidate solution X_{new} is always accepted if $c(X_{new}) < c(X_{current})$, and accepted with probability $e^{-(c(X_{new})-c(X_{current}))/Temp}$ if $c(X_{new}) > c(X_{current})$, where $Temp$ is the *temperature*. The temperature is initially set at $c(X_{init})P_{init}$ where $c(X_{init})$ is the objective function value of the initial solution X_{init} and P_{init} is an initialization constant. The current

temperature is gradually decreased during the course of the algorithm as *h.Temp*, where $0 < h < 1$ is a fixed parameter. The algorithm returns the best found solution after a fixed number of iterations.

4.1. Optimization stage: Initial solution

This section provides the details regarding the creation of the initial solution. In order to generate an initial solution, we first discard all the routing elements from the original LNG-DIRP and solve the resulting assignment problem over the planning horizon to optimality. Depending on the IRP application the quality of this initial solution varies for example, if the transportation cost accounts for a small proportion of the total cost, then this initial solution is of high quality. However, in the ALNS, the quality of the initial solution is not a major concern since a powerful improvement algorithm can fast compensate for a low quality initial solution.

The objective function of the assignment problem includes all the costs except for the routing cost:

$$\text{minimize } TC_{assignment} = PC^P + PC^L + HC^L + HC^P, \tag{39}$$

subject to constraints (4)–(5), (9)–(10), (19)–(28), and (33)–(37).

In the next and last step of building up the initial solution, the quantities obtained from the assignment problem are used to construct the routes. To this end, for each time period we load the assigned quantities to the first vehicle in set $\mathcal{V}$ with respect to the vehicle capacity. If the capacity of the vehicle is reached and there are still quantities that are not loaded, then another vehicle is used. We repeat this assignment process until all the quantities in the time period have been loaded onto vehicles. The procedure is reiterated for all the time periods, which results in a feasible initial solution.

4.2. Metaheuristic stage: An adaptive large neighborhood search algorithm

Introducing inventories as a new element to a VRP adds a new dimension to the problem. Hence, relying on solely classical operators will hardly direct the algorithm towards a high quality solution. In order to enable the algorithm to make the necessary trade-offs between routing and inventory holding costs and properly search the feasible region, we need to develop operators that deal with inventory levels. Here, we borrow some classical removal and insertion operators from the literature. Moreover, we introduce a set of *break-move* operators to give the opportunity to the replenishments in this distribution network to take place in different quantities and time periods. In this algorithm, each visit is defined as a vector that contains information on the index of the filling station, time period of the delivery, quantity delivered, the number of time period of which the demand is covered by the visit, and vehicle used for the service.

4.2.1. Removal operators

We define ten removal operators ($R = \{1, \ldots, 10\}$) which initially have equal probabilities of being part of the destroy phase. For each removal operator, γ visits are removed and added to the *removal list* $\mathcal{L}$, defined as an ordered set. We note that $\gamma \in [\underline{\gamma}, \overline{\gamma}]$. The value of γ is defined at the start of each removal operator and may vary from one iteration to another. Algorithm 2 shows the steps of this procedure.

Algorithm 2: Generic structure of the removal procedure.

input : A feasible solution X, and maximal number of iterations γ
output: A partially destroyed solution X_p, and a list of removed visits $\mathcal{L}$

```
Initialize removal list (L ← ∅)
for γ iterations do
    Apply removal operator to find a visit to remove
    Add the chosen visit to the list L
    Remove the chosen visit from X
```

In the following sections, we describe the removal operators used in the ALNS.

1. **Random removal (RR):** This operator assigns a random value to γ, then randomly removes γ visits from the existing solution, and adds them to $\mathcal{L}$. The main advantage of this operator is to diversify the search.
2. **Worst-distance removal (WDR):** The operator randomly chooses an existing route and evaluates each filling station (customer) on the route based on the distance traversed between the customer and the preceding and following nodes. The algorithm, then, removes the furthest filling station on the route. This process is reiterated until γ visits have been removed from the current solution and added to $\mathcal{L}$. The goal of this operator is to give a chance to these visits to be inserted into closer routes.
3. **Route removal (ROR):** In this operator, the algorithm randomly determines the number of routes, ρ, to be removed. The ALNS repeatedly chooses routes and inserts all the visits on those routes in $\mathcal{L}$. This continues until ρ routes have been removed from the existing solution.

4. **Time period removal (TR):** The aim of this operator is to randomly choose several time periods, and remove the routes that take place in them. This provides an opportunity to move the removed visits to the next or the previous time periods and combine them with other visits to the corresponding customers, and by this avoid large fixed costs of the routes in that specific time period. The operator selects a random number τ. It then randomly chooses τ time periods out of the set of time periods containing routes, and destroys all the routes in those time periods. In the next step, it inserts all the removed visits in $\mathcal{L}$. This operator is applied until τ time periods have been removed.
5. **Worst-vehicle capacity utilization removal (WVR):** This operator calculates the percentage capacity utilization of each vehicle in the current solution. It then removes the ρ routes with the lowest utilization rate and inserts their visits in $\mathcal{L}$.
6. **Global smallest quantity removal (GSR):** In order to give the chance to smaller delivered quantities to be combined with larger ones, this operator identifies the smallest delivered quantity. It, then, removes it from its route and adds it to the removal list $\mathcal{L}$. The operator repeatedly performs this operation until γ visits have been removed.
7. **Global worst distance removal (GWR):** Similar to worst distance operator, this operator looks for those visits that are relatively far from other visits on the same route. The difference is that, in this operator all the visits in the existing solution (and not in a route) are first evaluated. The worst visit in terms of distance is then removed from the corresponding route, and added to the list $\mathcal{L}$.
8. **Shaw static removal (SR-I):** The aim of this operator, and the next two, is to remove a group of visits that are in a way related. This provides an opportunity to seek a better solution by inserting these visits in new routes. The operator first randomly chooses an existing visit and removes it from its route. The algorithm then compares all the other visits in the current solution with the removed visit and gives a score to each based on three different criteria: 1) Time period (the closer the visit is to the removed filling station in terms of time period, the higher the score is for visit, i.e., a visit that takes place in the same time period as the removed visit gets the highest score), 2) distance between the two (the closer they are, the higher the score is for the visit), and 3) route (if the visit belongs to the same route at the removed filling station, then the visit gets the score). After evaluating all the visits in the exiting solution, then γ visits with highest scores are removed from the solution and added to removal list $\mathcal{L}$.
9. **Shaw static multiple time period removal (SR-II):** This operator is similar to SR-I but with a wider range of criteria. More specifically, we also consider the ratio of the quantity to the total quantity loaded on the vehicle; closer ratios result in a higher score for a filling station. Another criterion is the ratio of the demand at the filling station to the quantity delivered in that visit. The next criterion is the mode of transportation for the removed visit and other visits. Visits by the same transportation mode will get a higher score.
10. **Shaw dynamic removal (SR-III):** Considering the three criteria explained in the previous operator, the Shaw dynamic removal evaluates all existing visits and chooses the one with highest score. It then removes that visit from the solution and adds it to $\mathcal{L}$. In the next step, it updates all the routes and starts evaluating all the visits based on the criteria. After obtaining the scores for all the visits, the operator removes the visit with the highest score. This continues until γ visits have been removed from the solution.

If an operator is chosen to modify number of routes and time periods, the algorithm continues to perform removals until γ visits have been removed. If it is impossible to remove exactly γ visits by removing routes or time periods, the algorithm removes a smaller number of visits closest to γ.

4.2.2. Break-move operators

We now propose and elaborate on four operators ($B = \{1, \ldots, 4\}$) that break the removed quantities and move them from their original time period. The algorithm starts from the first visit in $\mathcal{L}$ and randomly chooses one of the break-move operators to modify the visit. The algorithm then removes the visit (and other visits, if necessary) from $\mathcal{L}$ and adds the relevant modified visits to a modified removal list $\mathcal{L}'$. Similar to other operators in this ALNS, at the start, these operators have an equal probability of being chosen. In later iterations, these probabilities are updated depending on the past performance of the operators. Fig. 4(a) shows the inventory level at filling station i before the removal operators remove a visit to this filling station. As a result, a visit to the filling station in time period t is removed from its route. The destroyed solution is illustrated in Fig. 4(b). It should be noted that all quantities must be updated over time. For example, if quantity q at time t is moved backwards for, say two time periods, then the new quantity at time $t-2$ should be $q_{new} = q + q_{deterioration}$. This means that quantity q_{new} is delivered at time period $t-2$ and loses $q_{deterioration}$ over two time periods. This has been incorporated into the heuristic, but in order to avoid a lengthy explanation, this is not mentioned in the formal description. The general structure of the break-move procedure is presented in Algorithm 3 .

1. **No break-move (NBM):** The algorithm gives the chance to a removed visit to remain unchanged in terms of the delivery time and quantity, see Fig. 4(c). This visit could be conducted by the same vehicle or a new one. If this operator is chosen then the visit is moved from $\mathcal{L}$ to $\mathcal{L}'$.
2. **No break with backward move (NBB):** This operator moves a removed quantity to an earlier time period without breaking it. Each filling station can be replenished at most once during a time period. Therefore, if a quantity of q_c that was supposed to be delivered to filling station i in time period t_c needs to be moved backwards, then the algorithm checks two conditions about previous time periods: 1) whether there is a stock-out situation in period $t_c - 1$ for filling station i. This happens in a destroyed solution when the quantity, e.g. q_p, that fulfills the demand of period $t_c - 1$ is already removed from its route. It should be noted that the original delivery time of quantity q_p (say t_p) depends on how many

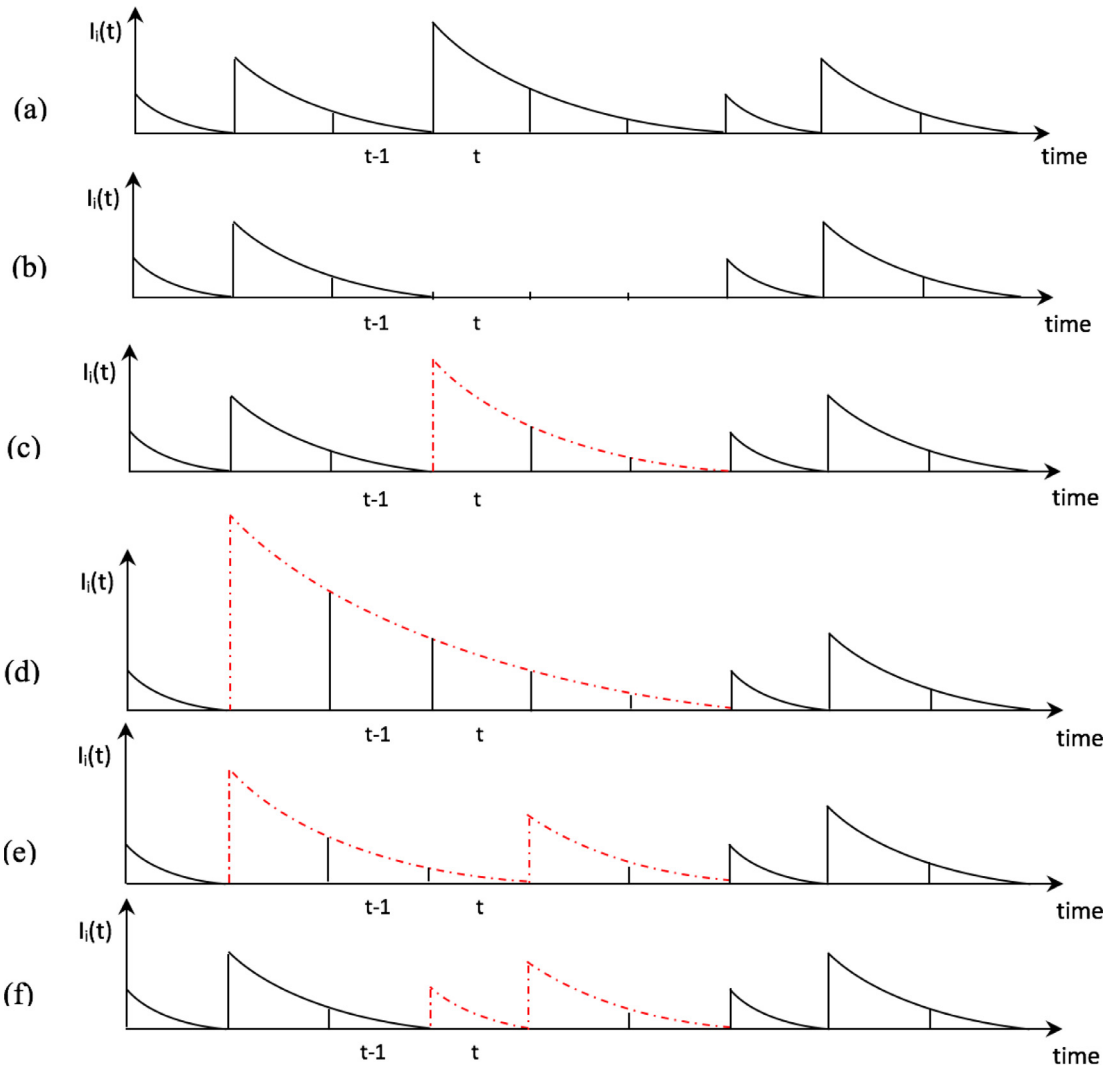

Fig. 4. The effects of break-move operators on a destroyed solution.

periods it was supposed to cover. In this case, both q_c and q_p are removed from $\mathcal{L}$ and the combined quantity $q_c + q_p$ with delivery time of t_p is added to $\mathcal{L}'$. 2) Whether the inventory level at the filling station during period $t_c - 1$ is positive. Then the algorithm rolls back to the time period, say t_p, in which the previous replenishment, q_p, of the filling station was made. In the next step, the ALNS removes q_p from its route and q from $\mathcal{L}$, and adds a quantity of $q_c + q_p$ with due date of t_p to the modified removal list $\mathcal{L}'$. Fig. 4(d) illustrates the resulting inventory level for the filling station.

3. **Break with backward move (BBM):** This operator aims at breaking quantities based on the demand during one time period. Operators that involve breaking a quantity are applicable to those removed quantities that cover the demand of more than one time period. Consider a quantity q_c in $\mathcal{L}$ that was supposed to be delivered at time period t_c to cover the demand of τ periods. If this operator applies to q_c, then part of the quantity, say q_1, that is sufficient to fulfill the demand of time period t_c is moved backwards (see *NBB* operator). The rest of the quantity $(q_c - q_1)$ is used to cover demand arising during a time interval starting at $t_c + 1$. The delivery of this quantity is therefore postponed to $t_c + 1$, and a quantity of $q_c - q_1$ with delivery date $t_c + 1$ is added to $\mathcal{L}'$. Fig. 4(e) depicts how this operator would change the visits to this filling station, hence the inventory level over different time periods.
4. **Break with forward move (BFM):** Similar to the *BBM* operator, this operator also splits a quantity q_c and adds the quantity $q_c - q_1$ to $\mathcal{L}'$. In this part, however, we do not move the quantity q_1 backwards. This means that q_1 with delivery date t_c is also added to $\mathcal{L}'$, see Fig. 4(f).

4.2.3. Insertion operators

In the matheuristic developed in this paper, we use five different insertion operators ($I = \{1, \ldots, 5\}$) which, if chosen, take quantities from $\mathcal{L}'$ and add them into different routes in their best position, see Algorithm 4 .

Algorithm 3: Generic structure of the break-move procedure.

input : The removal list $\mathcal{L}$
output: A modified list of removed visits $\mathcal{L}^{'}$

Let V_m be the largest vehicle capacity in the fleet
for all the visits in $\mathcal{L}$ **do**
  Let i be the filling station, and t_c and q_c be the time and quantity of the current visit, respectively
  Select a break-move operator
  if NBM is chosen **then**
    Add the current visit to $\mathcal{L}^{'}$ without any modifications
  if NBB is chosen **then**
    if $t > 0$ **and** the stock level at filling station i at $t-1$ is negative **then**
      Find the previous visit to the filling station which is in $\mathcal{L}$
      Let t_p and q_p be the time and quantity of that visit, respectively
      if $q_c + q_p \leq V_m$ **then**
        Add visit ($t = t_p$, $q = q_c + q_p$) to $\mathcal{L}^{'}$ and remove visit ($t = t_p$, $q = q_p$) from $\mathcal{L}$
      else
        Add the current visit to $\mathcal{L}^{'}$ without any modifications
    if $t > 0$ **and** the stock level at filling station i at $t-1$ is non-negative **then**
      Find the previous visit to the filling station, ($t = t_p$, $q = q_p$), in an existing route
      if $q_c + q_p \leq V_m$ **then**
        Add visit ($t = t_p$, $q = q_c + q_p$) to $\mathcal{L}^{'}$ and remove visit ($t = t_p$, $q = q_p$) from its route
      else
        Add the current visit to $\mathcal{L}^{'}$ without any modifications
    else
      Add the current visit to $\mathcal{L}^{'}$ without any modifications
  if BBM is chosen **then**
    if τ (the number of periods that are covered by q_c) > 1 **then**
      Let q_1 be the quantity that covers the demand of period t_c
      if NBB could be implemented on (t_c, q_1) **then**
        Apply NBB to (t_c, q_1) and add $(t_c + 1, q_c - q_1)$ to $\mathcal{L}^{'}$
      else
        Add the current visit to $\mathcal{L}^{'}$ without any modifications
    else
      Add the current visit to $\mathcal{L}^{'}$ without any modifications
  if BFM is chosen **then**
    if $\tau > 1$ **then**
      Let q_1 be the quantity that covers the demand of period t_c
      Add visits (t_c, q_1) and $(t_c + 1, q_c - q_1)$ to $\mathcal{L}^{'}$
    else
      Add the current visit to $\mathcal{L}^{'}$ without any modifications

Algorithm 4: Generic structure of the insertion procedure.

input : The modified removal list $\mathcal{L}^{'}$
output: X_{new}

for all visits in $\mathcal{L}^{'}$ **do**
  Add the visit to the destroyed solution using the insertion operators (I)

1. **Greedy insertion (GI):** This operator inserts a quantity into different existing routes in a specific time period. It also checks possible new routes with single visit in that time period. It finally assigns the quantity to the best possible position. To this end, it chooses the position that increases the total cost of the destroyed solution the least. In another words, suppose that there is a visit with quantity q to filling station i in $\mathcal{L}^{'}$. Assume that set $\mathcal{P}$ includes all the possible positions to insert this visit in the existing destroyed solution. The total cost of the destroyed solution before and after inserting quantity q to position p ($p \in \mathcal{P}$) are denoted by TC_b^p and TC_a^p, respectively. This operator chooses p^* where $p^* = \arg\min_{p\in\mathcal{P}}\{TC_a^p - TC_b^p\}$.
2. **Greedy insertion with noise function (GIN):** Since choosing the best possible position for a visit (as in the GI operator) does not guarantee moving towards the best possible solution, the GIN operator gives a chance to other positions to be chosen for this visit. To this end, it evaluates all possible positions and selects $p^* = \arg\min_{p\in\mathcal{P}}\{TC_a^p - TC_b^p + \overline{C^T}\mu\epsilon\}$, where $\overline{C^T}$ is the maximum cost of transportation between nodes in the network, μ is the noise parameter which we set equal to 0.1, and ϵ is a random number in $[-1, 1]$.
3. **Regret insertion (RI):** The GI operator assigns a visit to the best possible position in a destroyed solution without an evaluation of the consequences that this may bring to the assignment of the next visit in the list $\mathcal{L}^{'}$, i.e., it may be possible that the next visit in the list will have very costly positions as options. To give the opportunity to the algorithm to get around such situations, a "regret insertion" operator is introduced. The operator goes through the modified removal list and for visit $\chi \in \mathcal{L}^{'}$ with quantity q finds the best and second best possible position; p_q^1 and p_q^2 where $p_q^1 = \arg\min_{p\in\mathcal{P}}\{TC_a^p - TC_b^p\}$ and $p_q^2 = \arg\min_{p\in\mathcal{P}\setminus p_q^1}\{TC_a^p - TC_b^p\}$. The operator calculates the increase in the total cost function when assigning the visit to the best possible position, Δ_χ^1, and the second best position, Δ_χ^2. Finally, the operator finds $\chi^* = \arg\max_{\chi\in\mathcal{L}'}\{\Delta_\chi^2 - \Delta_\chi^1\}$ in $\mathcal{L}^{'}$ and inserts it in the best possible position.
4. **Regret insertion with noise function (RIN):** The RIN operator includes a random element into the RI operator so that the visit with the highest value between the second and first possible positions ($\Delta_\chi^2 - \Delta_\chi^1$) in $\mathcal{L}^{'}$ is not necessarily the first visit to be inserted into the destroyed solution. The structure of the operator is similar to that of RI where $\chi^* = \arg\max_{\chi\in\mathcal{L}'}\{\Delta_\chi^2 - \Delta_\chi^1 + \overline{C^T}\mu\epsilon\}$
5. **Biased greedy insertion (BGI):** Since we consider a heterogeneous fleet of vehicles that may vary in terms of fixed and variable transportation cost, all the above insertion operators look for a proper position in the destroyed solution to insert a visit from the list $\mathcal{L}^{'}$. If there is not enough capacity in the existing routes, the operators consider creating a new route using one of the idle vehicles. Given the large fixed cost for the larger vehicles, these may have only a very limited chance of being chosen for a new route, although they may be a potentially good option compared with the smaller vehicles. Indeed, the algorithm may be able to later add more quantities to these larger vehicles and eventually finds a very good solution. When there is a need for a new route, this operator randomly chooses a vehicle type to initialize a new route.

After all visits in $\mathcal{L}^{'}$ have been inserted in the destroyed solution, we conduct a neighborhood search on the resulting solution. To this end, for each route in the solution, we find the optimal order of the filling stations using a tabu search algorithm. The algorithm considers a route and performs several iterations. The process swaps each pair of filling stations (if the pair is not on the tabu list) on a route if this results in a better total cost. The pair is then added to the tabu list and will not be swapped for a given number of iterations.

4.2.4. Adaptive selection of operators

The selection of the removal, break-move, and insertion operators is based on a roulette-wheel mechanism. At the first iteration all operators of a given type have an equal probability of being selected. This means the ten removal, four break-move, and five insertion operators, have an initial probability of 1/10, 1/4, and 1/5, respectively. As the algorithm proceeds, at every N_w iterations, these probabilities are updated to $P_o^{t+1} = P_o^t\,(1 - r_p) + r_p\,\pi_i/\omega_i$, where $o \in (R \cup B \cup I)$, r_p is the roulette wheel parameter, π_i is the score of operator i, and ω_i is the number of times it was used during the last N_w iterations. The scores of the operators reflect their performance over the course of the previous N_w iterations. If at an iteration a new best solution is found, the scores of all three operators chosen in that iteration (removal, break-move, and insertion) are incremented by σ_1; if the solution is better than the current solution (and not the best solution), they are increased by σ_2; and if the solution is worse than the current solution but accepted, the scores are increased by σ_3.

5. Computational results

We now present the computational experiments performed to evaluate the quality of our matheuristic. We have generated instances partially based on the data from the Netherlands. Since LNG distribution networks are not yet well-established and the technical aspects are in an evolving phase, some assumptions (e.g., cost parameters, evaporation pattern at the filling stations, the dynamics of the LNG price and demand at the retail market are made based on existing practices, and after consulting with the industrial experts involved in the project funded by DINALOG (2018).

The generated data set contains 60 instances, partitioned into four sets of 15 instances, such that the instances in each set are similar in terms of the number of filling stations and the time periods. Each instance comes with the following information: (*i*) Number of filling stations and number of time periods, (*ii*) distances between each pair of nodes (these are

Table 1
Parameters used for the experiment.

Category	Notation	Description	Typical values
Filling station i	$\bar{S}_i$	Capacity, m^3	500–1,500
	$\underline{S}_i$	Minimum allowed inventory, m^3	0
	S_i	Initial inventory, m^3	0
	D_{it}	Demand during period t, m^3/day	80–250
	C_i^F	Fixed ordering cost, € /order	70–130
	C_i^H	Unit holding cost per period, € /day	0.8–1.4
	θ	Deterioration rate, %	0.5
Vehicle v	$\bar{V}_v$	Capacity (truck), m^3	250–2,000
		Capacity (barge), m^3	2,000–7,500
	C_v^{FV}	Fix delivery cost (truck), € /order	100–300
		Fix delivery cost (barge), € /order	200–800
	C_v	Variable transportation cost (truck), € /km	0.7–1.0
		Variable transportation cost (barge), € /km	0.2–0.4
Storage facility	$\bar{S}_0$	Capacity (depending on $\mathcal{N}^L$), m^3	2,000–500,000
	$\underline{S}_0$	Minimum allowed inventory, m^3	$0.005\bar{S}_0$
	S_0	Initial inventory, m^3	$0.005\bar{S}_0$
	C_0^F	Fixed ordering cost, € /order	800–1,500
	C_0^H	Unit holding cost per period, € /day	0.2–0.6
	C_{0t}^P	Unit purchasing cost in period t, € /m^3	0.7–1.1
	θ	Deterioration rate, %	0.5

Table 2
Parameters used in the algorithm.

Category	Notation	Description	Typical Value
(i)	N_i	Total number of iterations	5,000
	N_W	Number of iterations for roulette wheel	400
	r_p	Roulette wheel parameter	0.75
	σ_1	Value added to the score when a new best solution is found	15
	σ_2	Value added to the score when a new better solution is found	5
	σ_3	Value added to the score when a new worse solution is found	15
(ii)	P_{init}	Startup temperature parameter	0.1
	h	Cooling rate	0.99
(iii)	$\underline{\gamma}$	Minimum number of nodes to remove	$0.05\lvert\mathcal{N}_{ret}\rvert$
	$\overline{\gamma}$	Maximum number of nodes to remove	$0.25\lvert\mathcal{N}_{ret}\rvert$
	μ	Noise parameter	0.1

based on actual geographical distances between randomly selected cities in the Netherlands), (*iii*) demand at each filling station at each time period, (*iv*) detailed information on each filling station including capacity and cost parameters, (*v*) detailed information on storage facility including capacity and cost parameters, (*vi*) detailed information on each type of vehicles including capacity and cost parameters, see Table 1. It should be noted that each instance contains at least one from each vehicle. The information is based on estimates provided by practitioners in the field. Instances are identified as *NL_F_P*, where *NL* stands for the Netherlands, *F* is the number of filling stations, and *P* is the number of time periods. The algorithm was coded in Java on a 64-bit machine equipped with an *i*7-6700K processor and 24GB of RAM. The following subsections provide the experiment design and results.

5.1. Parameter tuning and analysis

The implementation of the ALNS heuristic contains 11 user-controlled parameters which are provided in Table 2, along with the value we used in our numerical experiments. As in Demir et al. (2012), we divide the parameters into three categories. Category (*i*) includes the parameters that control the overall algorithm and selection of operators. Category (*ii*) includes the parameters that control the master search framework (i.e., simulated annealing). Category (*iii*) includes parameters used by operators.

In order to set the value of these parameters, we conduct several numerical experiments to tune the parameters. All values presented in Table 2 are the results of tests performed on a tuning set. We considered multiple sets of values and for each parameter setting we applied our ALNS to all instances from the tuning set five times. To ensure comparison fairness, we used the same initial solution and seeds for the random numbers across runs with different parameter values.

5.1.1. Tuning of the roulette wheel mechanism parameters

In order to tune the control parameters σ_1, σ_2, and σ_3 used to adjust the scores, we ran the numerical tests on 15 instances. We considered 10 different combinations for $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)$: (15,10,5), (15,5,10), (10,10,10), (5,15,10), (5,10,15), (5,15,15), (15,5,15), (15,15,5), (10,15,5), and (10,5,15). For the rest of parameters, we aligned the values with what used by

Table 3
Tuning of the roulette wheel mechanism parameters (σ_1, σ_2, σ_3).

	$\sigma = (15, 5, 10)$		$\sigma = (10, 10, 10)$		$\sigma = (5, 15, 10)$		$\sigma = (5, 10, 15)$		$\sigma = (10, 15, 5)$	
Instance	*Best*	*%Dev*	*Best*	*%Dev*	*Best*	*%Dev*	*Best*	*%Dev*	*Best*	*%Dev*
NL_7_4	14,364.73	0.14	14,364.73	0.14	14,364.73	0.14	14,364.73	0.14	14,364.73	0.14
NL_10_5	31,110.89	0.02	31,410.89	0.98	31,382.35	0.89	31,421.46	1.02	31,317.49	0.68
NL_20_3	38,842.63	0.88	38,599.43	0.25	38,754.17	0.65	38,873.92	0.96	38,854.74	0.91
NL_20_5	57,457.13	0.75	58,130.48	1.93	57,670.23	1.12	57,838.78	1.42	57,513.21	0.85
NL_30_3	49,048.89	0.98	48,891.36	0.66	48,878.56	0.63	48,822.47	0.52	48,977.70	0.84
NL_30_5	86,478.52	0.89	86,669.61	1.11	86,729.43	1.18	86,546.78	0.97	86,663.05	1.10
NL_40_2	44,372.98	0.06	44,465.98	0.27	44,558.98	0.48	44,463.98	0.26	44,353.98	0.01
NL_40_3	69,124.11	0.60	69,824.88	1.62	69,744.60	1.51	69,352.18	0.94	69,425.98	1.04
NL_40_5	105,412.65	1.02	105,354.12	0.96	105,258.49	0.87	105,128.44	0.74	105,221.69	0.83
NL_50_3	83,795.18	0.65	83,845.37	0.71	84,025.21	0.93	83,784.55	0.64	83,724.58	0.56
NL_60_3	101,422.10	0.84	101,561.10	0.98	102,821.10	2.23	102,224.10	1.64	102,846.10	2.26
NL_70_2	71,564.28	0.36	71,579.28	0.38	72,418.28	1.55	73,417.28	2.95	72,062.28	1.05
NL_80_2	90,756.93	0.58	90,706.93	0.52	92,272.93	2.26	92,146.93	2.12	91,321.93	1.20
NL_90_3	160,263.07	0.16	160,193.07	0.11	161,643.07	1.02	161,286.07	0.79	161,690.07	1.05
NL_100_2	102,146.29	0.11	102,277.29	0.24	103,347.29	1.29	102,904.29	0.85	104,465.29	2.38
Average		0.54		0.72		1.12		1.06		0.99

Table 4
Tuning of the roulette wheel mechanism parameters (σ_1, σ_2, σ_3).

	$\sigma = (15, 10, 5)$		$\sigma = (5, 15, 15)$		$\sigma = (15, 5, 15)$		$\sigma = (15, 15, 5)$		$\sigma = (10, 5, 15)$	
Instance	*Best*	*%Dev*	*Best*	*%Dev*	*Best*	*%Dev*	*Best*	*%Dev*	*Best*	*%Dev*
NL_7_4	14,344.39	0.00	14,364.73	0.14	14,364.73	0.14	14,344.39	0.00	14,364.73	0.14
NL_10_5	31,105.49	0.00	31,122.59	0.05	31,110.89	0.02	31,110.89	0.02	31,110.89	0.02
NL_20_3	38,629.39	0.33	38,650.64	0.38	38,504.13	0.00	38,860.19	0.92	38,751.91	0.64
NL_20_5	58,342.14	0.00	58,374.34	0.05	58,386.14	0.08	58,494.93	0.26	58,458.02	0.20
NL_30_3	48,887.86	0.65	48,770.27	0.41	48,570.98	0.00	48,673.57	0.21	48,835.55	0.54
NL_30_5	87,047.31	0.15	86,999.93	0.10	86,916.69	0.00	86,949.12	0.04	87,004.61	0.10
NL_40_2	44,359.98	0.03	44,357.98	0.02	44,373.98	0.06	44,351.98	0.01	44,347.98	0.00
NL_40_3	69,418.67	1.03	68,987.70	0.40	68,709.51	0.00	68,732.05	0.03	68,866.58	0.23
NL_40_5	106,005.72	0.08	105,973.41	0.05	105,925.66	0.00	105,980.69	0.05	106,035.76	0.10
NL_50_3	83,513.37	0.01	83,609.30	0.13	83,504.63	0.00	83,545.85	0.05	83,601.43	0.12
NL_60_3	101,331.10	0.75	101,087.66	0.51	101,476.41	0.90	101,523.97	0.95	100,573.38	0.00
NL_70_2	71,355.28	0.06	71,343.38	0.05	71,334.98	0.03	71,310.48	0.00	71,310.48	0.00
NL_80_2	90,602.93	0.41	90,583.33	0.39	90,235.43	0.00	90,589.63	0.39	90,572.83	0.37
NL_90_3	160,032.07	0.01	160,279.08	0.17	160,023.39	0.01	160,014.93	0.00	160,093.79	0.05
NL_100_2	102,090.29	0.06	102,110.99	0.08	102,032.69	0.00	102,046.19	0.01	102,071.39	0.04
Average		0.24		0.19		0.08		0.20		0.17

Demir et al. (2012), i.e. $h = 0.999$, $r_p = 0.1$, and we consider the interval $[0.10|\mathcal{N}_{ret}|, 0.25|\mathcal{N}_{ret}|]$ for the number of removable filling station in each iteration. We, then, ran the proposed algorithm 10 times on each instance for each combination of (σ_1, σ_2, σ_3). We report the worst and best five combinations in Tables 3 and 4, respectively. For each instance presented in these tables, the columns *%Dev* report the percentage of deviation from the best solution value calculated across all the five combinations in each table.

The results presented in Table 4 show that, considering the percentage of deviation, the combination (15,5,15) outperforms the other combinations on average. We, therefore, adopt this set of values for (σ_1, σ_2, σ_3) for the rest of our numerical analysis.

5.1.2. Tuning the roulette wheel control and the cooling rate parameters

In this section, we jointly tune the cooling rate parameter h, and the roulette wheel control parameter r_p. To this end, we set h at two levels, namely: 0.95 and 0.99, and we set r_p at 0.1, 0.25, 0.5, and 0.75. We then run a few larger instances (*NL*_50_3, *NL*_60_3, and *NL*_100_3) using the eight resulting combinations of these two parameters. Fig. 5 shows how the total cost of instance *NL*_100_3 changes over the iterations. In this figure, we report the result of the four combinations of (h, r_p) with the best performance. The combination (0.99, 0.75) gives a better solution over time. We, therefore, use these values for the rest of the experiments.

5.1.3. Tuning the parameters that specify the fraction of filling stations to be removed

In this problem, there are $|\mathcal{N}^L|$ filling stations served over $|\mathcal{T}|$ periods. At each iteration, we randomly remove a number of filling stations from a solution, as a percentage of the number of visits in the solution, $|\mathcal{N}_{ret}|$. In order to find an appropriate interval for the removal, we set the minimum and maximum on different levels and ran the algorithm on the instance *NL*_100_3. This analysis, summarized in Table 5, shows that the interval $[0.05|\mathcal{N}_{ret}|, 0.25|\mathcal{N}_{ret}|]$ for the number of

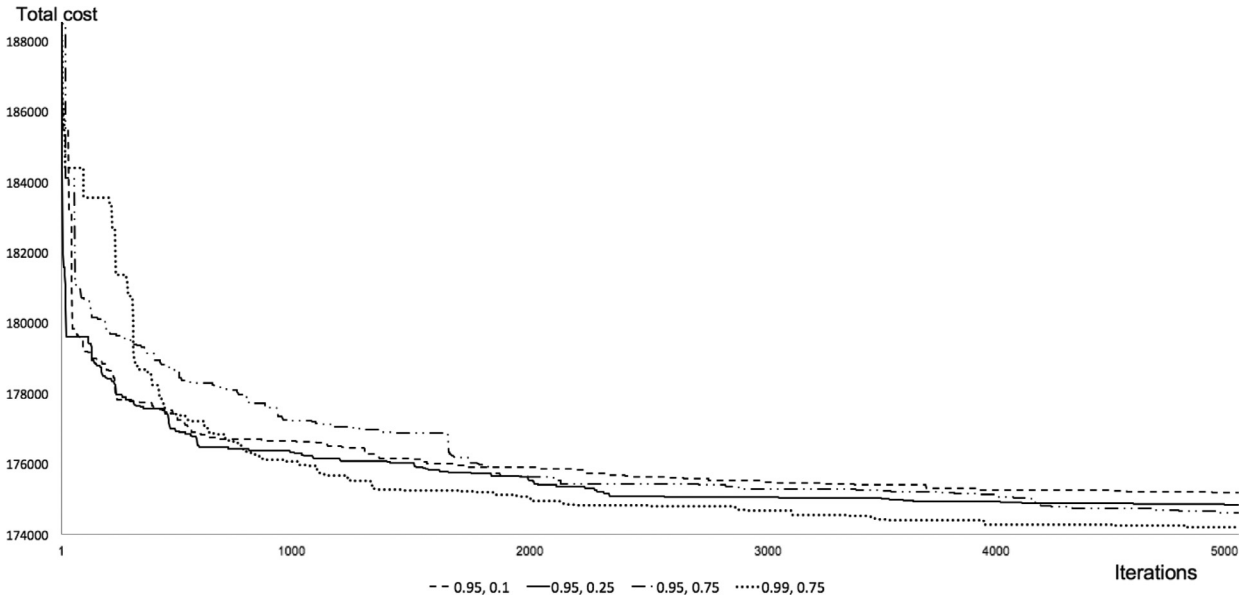

Fig. 5. Change in the total cost over iterations for a different set of combinations of r_p and h.

Table 5
The relative increase in the total cost function for various $\underline{\gamma}$ and $\overline{\gamma}$ parameters.

Upper bound % of $\lvert\mathcal{N}_{ret}\rvert$	Lower bound % of $\lvert\mathcal{N}_{ret}\rvert$ 5	10	15	20
10	0.02	–	–	–
15	0.31	0.33	–	–
20	0.17	0.51	0.50	–
25	**0.00**	0.28	0.39	0.27
30	0.18	0.28	0.21	0.09

filling stations to remove, yields a better solution. For each combination of lower and upper bounds, the table presents the deviation from the best solution found.

5.2. Performance comparison of the operators

Tables 6, 7, and 8 provide statistics on the percentage of the number of iterations and of the computing time for which each removal, break-move, and insertion operator was called. These results are obtained using one instance from each set identified for parameter-tuning.

Table 6 indicates that the frequency of using the removal operators do not significantly vary from one operator to another. Moreover, the table indicates that the removal operators are quite fast, hence can effectively be used for the rest of experiments. Table 7 shows that the NBM and NBB operators are used the most, and the NBB and BFM operators are used the least in these experiments. Since all four operators contribute for a better solution, we keep them as part of the algorithm for the rest of our experiment. As for the insertion operators, the GIN and BGI are used less frequently while RI and RIN on average are called more often, hence have higher contribution in the final solution, see Table 8. This table also shows that the latter two operators consume significantly more time than the rest of the operators.

5.3. Comparison between the performance of the matheuristic and that of CPLEX on the mathematical model

In this section we analyze the speed performance of our algorithm using small instances from the tuning set. The columns entitled “Best value” and “Avg value” present the best and average values of each instance over five runs, respectively. The column entitled “Avg CPU time” reports the average CPU time in minutes. Table 9 shows that the proposed matheuristic performs well, compared to applying CPLEX to the mathematical formulation. Since the problem is NP-hard, it is only possible to solve small-size instances (i.e., up to 10 filling stations and two time periods). We note that CPLEX was

Table 6
Removal operators' contribution as a percentage in 5,000 iterations (and the CPU time spent).

	Removal operators									
Instance	RR	WDR	ROR	TR	WVR	GSR	GWR	SR-I	SR-II	SR-III
NL_7_4	0.9 (0.0)	0.9 (0.0)	11.3 (0.0)	10.9 (0.0)	9.9 (0.0)	13.8 (0.0)	13.4 (0.0)	13.3 (0.0)	12.8 (0.0)	12.8 (0.0)
NL_10_5	0.9 (0.0)	0.9 (0.0)	11.7 (0.0)	11.9 (0.0)	13.3 (0.0)	10.6 (0.0)	12.8 (0.0)	12.4 (0.0)	13.2 (0.0)	12.2 (0.0)
NL_20_3	0.7 (0.0)	0.6 (0.0)	12.2 (0.0)	10.9 (0.0)	10.3 (0.0)	15.6 (0.0)	11.2 (0.0)	13.0 (0.0)	12.5 (0.0)	13.0 (0.0)
NL_20_5	0.9 (0.0)	0.8 (0.0)	13.2 (0.0)	11.5 (0.0)	9.7 (0.0)	10.9 (0.0)	14.0 (0.0)	12.6 (0.0)	12.8 (0.0)	11.6 (0.0)
NL_30_3	0.6 (0.0)	0.6 (0.0)	12.0 (0.0)	12.3 (0.0)	10.0 (0.0)	14.5 (0.0)	11.1 (0.0)	11.8 (0.0)	14.9 (0.0)	12.2 (0.0)
NL_30_5	0.9 (0.0)	0.7 (0.0)	11.2 (0.0)	8.7 (0.0)	11.5 (0.0)	14.9 (0.0)	10.5 (0.0)	13.7 (0.0)	13.3 (0.0)	14.6 (0.0)
NL_40_2	0.8 (0.0)	0.7 (0.0)	14.1 (0.0)	9.6 (0.0)	16.2 (0.0)	11.1 (0.0)	11.5 (0.0)	11.5 (0.0)	12.9 (0.0)	11.6 (0.0)
NL_40_3	0.7 (0.0)	0.7 (0.0)	10.9 (0.0)	10.2 (0.0)	11.5 (0.0)	13.4 (0.0)	11.9 (0.0)	10.6 (0.0)	14.0 (0.0)	16.1 (0.0)
NL_40_5	0.6 (0.0)	0.7 (0.0)	11.6 (0.0)	10.5 (0.0)	10.2 (0.0)	15.7 (0.0)	10.5 (0.0)	11.9 (0.0)	13.8 (0.0)	14.5 (0.0)
NL_50_3	0.7 (0.0)	0.8 (0.0)	13.3 (0.0)	10.9 (0.0)	12.2 (0.0)	12.5 (0.0)	10.4 (0.0)	12.6 (0.0)	14.5 (0.0)	12.1 (0.0)
NL_60_3	0.7 (0.0)	0.2 (0.0)	12.9 (0.0)	11.5 (0.0)	10.6 (0.0)	12.3 (0.0)	12.6 (0.0)	12.5 (0.0)	14.5 (0.0)	12.2 (0.0)
NL_70_2	0.8 (0.0)	0.9 (0.0)	12.2 (0.0)	9.8 (0.0)	15.5 (0.0)	13.0 (0.0)	12.1 (0.0)	11.6 (0.0)	12.7 (0.0)	11.4 (0.0)
NL_80_2	0.7 (0.0)	0.7 (0.0)	14.5 (0.0)	9.5 (0.0)	14.5 (0.0)	15.1 (0.0)	11.0 (0.0)	11.2 (0.0)	12.0 (0.0)	10.8 (0.0)
NL_90_3	0.4 (0.0)	0.2 (0.0)	12.7 (0.0)	11.9 (0.0)	14.1 (0.0)	11.3 (0.0)	11.2 (0.0)	13.6 (0.0)	12.2 (0.0)	12.4 (0.0)
NL_100_2	0.7 (0.0)	0.7 (0.0)	15.5 (0.0)	9.9 (0.0)	14.2 (0.0)	11.7 (0.0)	9.9 (0.0)	11.0 (0.0)	12.1 (0.0)	14.3 (0.0)

Table 7
Break-move operators' contribution as a percentage in 5,000 iterations (and the CPU time spent).

	Break-move operators			
Instance	NBM	NBB	BBM	BFM
NL_7_4	31.7 (0.0)	12.0 (0.0)	30.4 (0.0)	25.9 (0.0)
NL_10_5	46.0 (0.0)	16.9 (0.1)	25.5 (0.0)	11.6 (0.0)
NL_20_3	45.2 (0.0)	16.6 (0.0)	19.4 (0.0)	18.8 (0.0)
NL_20_5	48.7 (0.0)	21.9 (0.0)	23.6 (0.0)	5.8 (0.0)
NL_30_3	41.6 (0.0)	18.3 (0.0)	19.8 (0.0)	20.3 (0.0)
NL_30_5	48.3 (0.0)	6.3 (0.0)	24.9 (0.0)	20.5 (0.0)
NL_40_2	31.5 (0.0)	30.6 (0.0)	33.8 (0.0)	4.1 (0.0)
NL_40_3	42.9 (0.0)	17.5 (0.0)	22.4 (0.0)	17.2 (0.0)
NL_40_5	46.1 (0.0)	6.6 (0.0)	27.5 (0.0)	19.8 (0.0)
NL_50_3	34.4 (0.0)	25.6 (0.0)	34.3 (0.0)	5.7 (0.0)
NL_60_3	50.7 (0.0)	14.7 (0.0)	27.3 (0.0)	7.3 (0.0)
NL_70_2	32.8 (0.0)	31.8 (0.0)	30.4 (0.0)	5.0 (0.0)
NL_80_2	30.6 (0.0)	31.1 (0.0)	33.2 (0.0)	5.1 (0.0)
NL_90_3	44.6 (0.0)	7.8 (0.0)	37.6 (0.0)	10.0 (0.0)
NL_100_2	30.6 (0.0)	32.3 (0.0)	32.6 (0.0)	4.5 (0.0)

Table 8
Insertion operators' contribution as a percentage in 5,000 iterations (and the CPU time spent).

	Insertion operators				
Instance	GI	GIN	BGI	RI	RIN
NL_7_4	20.0 (0.0)	18.4 (0.0)	20.8 (0.0)	20.9 (0.0)	19.9 (0.0)
NL_10_5	21.0 (0.0)	18.9 (0.0)	15.1 (0.0)	22.7 (0.0)	22.3 (0.0)
NL_20_3	19.0 (0.0)	17.9 (0.0)	17.5 (0.0)	22.5 (0.0)	23.1 (0.0)
NL_20_5	20.2 (0.0)	20.2 (0.0)	16.5 (0.0)	20.0 (0.0)	23.1 (0.0)
NL_30_3	21.4 (0.0)	18.9 (0.0)	17.0 (0.0)	23.1 (0.0)	20.0 (0.0)
NL_30_5	19.6 (0.0)	16.6 (0.0)	16.7 (0.0)	23.8 (0.0)	23.3 (0.0)
NL_40_2	22.9 (0.0)	18.0 (0.0)	12.1 (0.0)	23.5 (0.0)	23.5 (0.0)
NL_40_3	17.3 (0.0)	15.5 (0.0)	15.7 (0.0)	25.2 (0.2)	26.3 (0.2)
NL_40_5	20.3 (0.0)	17.1 (0.0)	15.5 (0.0)	23.4 (0.4)	23.7 (0.5)
NL_50_3	19.5 (0.0)	16.7 (0.0)	11.7 (0.0)	27.5 (0.0)	24.6 (0.0)
NL_60_3	16.2 (0.0)	9.0 (0.0)	17.7 (0.0)	28.6 (0.6)	28.4 (0.7)
NL_70_2	19.2 (0.0)	18.8 (0.0)	10.1 (0.0)	26.2 (1.3)	25.7 (1.2)
NL_80_2	21.7 (0.0)	17.0 (0.0)	12.3 (0.0)	24.0 (5.1)	25.0 (4.0)
NL_90_3	23.5 (0.0)	9.0 (0.0)	10.6 (0.0)	30.4 (6.6)	26.5 (4.9)
NL_100_2	21.1 (0.0)	20.1 (0.0)	10.6 (0.0)	24.4 (11.9)	23.8 (10.3)

Table 9
The comparison between the matheuristic and CPLEX for small-size instances.

Instance	ALNS (five runs)			CPLEX			
	Best value (€)	Avg value (€)	Avg CPU time (sec)	Best value (€)	CPU time (sec)	Gap (%)	δ(%)
NL_5_2	6,425	6,425	0.0	6,425	0.0	0.00	0.00
NL_5_3	8,917	8,917	0.0	8,917	0.0	0.00	0.00
NL_6_2	7,953	7,953	0.0	7,953	0.0	0.00	0.00
NL_6_4	15,336	15,392	0.0	15,336	3.0	0.00	0.00
NL_7_4	14,344	14,370	0.2	14,318	18.0	0.00	0.18
NL_8_2	9,154	9,189	0.0	9,154	5.0	0.00	0.00
NL_8_3	15,313	15,351	0.3	15,225	223.0	0.00	0.58
NL_8_4	18,027	18,337	0.0	17,918	3,398	0.00	0.61
NL_9_2	11,427	11,482	0.0	11,427	8.0	0.00	0.00
NL_9_3	16,862	16,862	0.5	16,916	3,600[a]	1.93	–
NL_9_4	22,252	22,345	0.5	22,217	212.2	0.00	0.16
NL_10_2	11,461	11,467	0.0	11,457	801.1	0.00	0.03
NL_10_5	31,395	32,226	0.8	37,112	3,600	19.18	–
NL_12_2	14,523	14,524	0.7	14,652	3,600	1.96	–
NL_12_4	32,565	32,791	0.8	35,831	3,600	15.91	–

[a] CPLEX cannot find a solution within 3600 seconds.

Table 10
Results of the matheuristic on the medium-size instances.

Instance	CPU time (sec)	Distance traveled (km)		Cost (€)					Utilization (%)		# of visits	
		Truck	Barge	Total	Purchasing	Routing	Filling stations	Storage facility	Truck	Barge	Truck	Barge
NL_9_7	0.9	6,833	0	40,078	9,375	11,676	12,897	6,130	87.7	–	42	0
NL_9_14	2.9	8,229	1,280	82,563	19,023	21,923	32,123	9,494	88.7	32.5	54	8
NL_9_30	8.2	1,829	10,581	167,243	40,911	28,621	77,169	20,542	65.7	72.3	8	97
NL_10_7	0.8	4,356	0	45,175	10,945	11,805	17,804	4,621	83.3	–	32	0
NL_10_14	1.9	7,623	1,709	82,647	19,769	19,222	32,615	11,041	89.3	48.9	55	14
NL_10_30	5.8	20,843	9,196	196,786	42,648	60,349	74,982	18,807	74.1	36.1	74	43
NL_15_7	2.1	2,654	1,194	60,241	15,522	15,236	23,138	6,345	80.0	93.5	20	37
NL_20_7	2.6	6,459	272	85,846	18,129	28,024	35,190	4,503	91.7	32.5	56	4
NL_20_14	7.5	15,352	1,053	175,969	38,054	52,038	72,686	13,191	90.9	80.5	97	16
NL_25_7	8.0	11,248	650	100,690	25,762	26,686	40,286	7,956	88.6	92.7	88	9
NL_30_7	10.8	10,331	2,610	116,861	26,952	31,288	52,897	5,724	88.1	97.4	50	39
NL_30_14	36.3	29,528	13,719	248,651	63,121	83,189	88,730	13,611	86.4	69.4	140	84
NL_35_7	13.9	12,591	4,910	141,204	35,137	46,985	52,783	6,299	88.5	75.3	85	48
NL_40_7	13.8	17,587	3,270	176,005	35,783	71,319	60,146	8,757	94.0	78.4	101	47
NL_40_14	110.4	41,846	0	338,304	84,027	101,041	121,858	31,378	91.5	-	371	0
Average									85.9	67.5		

run with a time limit of one hour. The results show that the matheuristic finds the same solutions as those of CPLEX, but much faster.

5.4. Results of the proposed algorithm on the medium- and large-size instances

To help assess the quality of the proposed algorithm, we present computational results in Tables 10 and 11 on medium- and large-size instances, respectively. We provide the best solution value, the CPU time, the total distance traveled with each vehicle, costs, utilization, and the number of visits performed by each type of vehicle. The columns “Truck” and “Barge”, under the title “Distance traveled”, give the total distance traveled with each type of vehicle. Under the column “Cost”, we list the total, purchase, and routing costs, and the costs incurred by the filling stations and the storage facility. In the columns “Truck” and “Barge” under the “Utilization” heading, we show the utilization rates of each type of vehicles. The last two columns provide the number of visits done by each type of vehicles.

5.5. Comparison of two replenishment policies with cost minimisation policy

Owing to the technological advancement, LNG is relatively safe to transport, see, e.g., Foss et al. (2003). In case of any accident, however, the consequences could be severe, see, e.g., Planas et al. (2015). This is the main reason why regulatory bodies may set strict rules on transportation routes and time windows to minimize in-transit quantities. In our study, apart from cost minimization policy, we analyze the effects of two policies that the operator of the inland distribution system may

Table 11
Results of the matheuristic on the large-size instances.

	CPU	Distance traveled		Cost (€)					Utilization		# of visits	
	time	(km)					Filling	Storage	(%)			
Instance	(sec)	Truck	Barge	Total	Purchasing	Routing	stations	facility	Truck	Barge	Truck	Barge
NL_50_7	236	14,308	1,126	192,166	50,251	59,173	73,107	9,635	91.5	59.4	185	16
NL_50_14	363	68,550	17,093	421,986	95,874	160,936	148,379	16,797	87.6	83.8	263	125
NL_50_30	2,230	115,368	33,642	868,958	201,438	305,718	322,868	38,933	87.8	72.6	565	230
NL_60_7	291	56,858	100,748	296,245	60,278	139,587	79,881	16,499	91.2	99.5	368	11
NL_60_14	1,128	105,317	6,438	468,871	112,115	163,479	173,848	19,429	91.9	92.1	535	100
NL_60_30	1,707	134,022	31,364	1,093,512	247,367	428,413	375,026	42,706	92.0	58.6	693	302
NL_70_7	143	22,816	1,145	311,793	73,097	96,891	122,077	19,728	93.1	86.3	228	19
NL_70_14	961	79,395	0	526,562	144,794	168,591	190,944	22,233	94.2	-	774	0
NL_70_30	3,385	224,374	44,285	1,326,662	304,852	527,737	448,569	45,504	88.9	87.4	702	407
NL_80_7	409	10,023	7,362	318,122	77,658	88,522	139,317	12,625	91.7	97.0	98	165
NL_80_14	665	13,053	15,063	701,717	160,476	181,650	331,631	27,960	89.8	93.9	96	302
NL_90_7	314	2,656	7,988	368,926	91,975	93,210	171,524	12,217	85.0	88.8	22	213
NL_90_14	1,015	68,155	51,942	713,095	182,109	232,802	274,254	23,930	93.2	90.7	318	314
NL_100_7	687	40,196	21,085	421,659	98,282	147,421	161,553	14,403	92.8	77.9	161	183
NL_100_14	2,947	684	31,962	737,943	204,193	187,962	321,218	24,570	98.9	90.8	9	753
Average									91.3	84.2		

Table 12
The effects of different replenishment policies.

		Dist. traveled		Cost (€)					Utilization			# of visits	
		(km)			Purch-		Filling	Storage	(%)				
Instance	Policy	Truck	Barge	Total	asing	Routing	stations	facility	Truck	Barge	FS	Truck	Barge
NL_20_14	CM	15,352	1,053	175,969	38,054	52,038	72,686	13,191	90.9	80.5	43.3	97	16
	FVL	13,822	994	202,121	39,041	40,000	107,569	15,511	90.3	96.2	73.8	60	13
	MQD	12,351	3,147	203,787	40,695	43,564	108,267	11,260	71.2	63.2	78.1	42	27
NL_30_14	CM	26,270	9,452	247,320	63,353	78,216	92,037	13,714	85.9	55.3	40.1	131	84
	FVL	26,310	1,771	294,368	65,638	51,740	154,978	22,012	84.4	87.7	72.8	98	23
	MQD	11,079	9,912	287,297	66,493	45,258	161,080	14,466	78.5	73.1	80.2	45	63
NL_40_7	CM	17,587	3,270	176,005	35,783	71,319	60,146	8,757	94.0	78.4	32.9	101	47
	FVL	11,582	1,171	199,511	34,314	40,527	112,322	12,348	85.1	98.1	63.7	56	21
	MQD	8,952	3,410	198,081	34,191	43,705	112,001	8,184	79.3	79.2	70.0	33	37
NL_50_7	CM	14,308	1,126	192,166	50,251	59,173	73,107	9,635	91.5	59.4	34.4	185	16
	FVL	6,997	2,608	237,410	54,529	36,087	137,159	9,635	90.2	93.4	68.5	62	31
	MQD	9,431	3,684	233,939	54,421	39,688	131,827	8,002	85.5	84.4	70.2	57	35

adopt. Moreover, the analysis gives insights into the performance of this distribution network in terms of capacity utilization, compared with sea shipping. The first policy consists of loading the vehicles as much as possible (full vehicle load, FVL). The second policy aims at delivering the largest possible quantity (maximum quantity delivery, MQD) when visiting a filling station. These considerations become highly important when there are regulations on large vehicles traveling to highly populated neighborhoods. In such situations, the fleet operator strives to keep the in-transit quantities or the number of visits to a minimum, compromising on the cost and keeping a minimum service level. Deviating from the cost minimization objective (CM–the benchmark policy) may also be preferable when a hazardous item like LNG is transported.

The effects of adopting FVL and MQD policies on the performance of the distribution system are studied on four instances, namely; *NL*_20_14, *NL*_30_14, *NL*_40_7, and *NL*_50_7, and the results are presented in Table 12. These policies do increase the overall cost of the distribution network, but yield a smaller number of visits, hence a lower routing cost compared with the benchmark policy (CM). In this table, we also report the average capacity utilization rate for trucks, barges, and filling stations (FS). It is observed that the FVL policy tends to maximally utilize the vehicles and the MQD policy delivers the maximum possible quantity to a filling station, which results in a higher utilization rate for the filling stations. In terms of inventory and delivery costs, FVL and MQD both result in higher filling station costs. This is due to the fact that these policies do not make trade-offs between inventory holding cost at the filling stations and other costs of the system including routing cost and holding cost at the storage facility. The MQD policy yields a lower cost at the storage facility, compared to the CM policy, since it shifts (pulls) the inventory towards the filling stations.

5.6. Analysis on the fleet size

To study the effects of the number of barges and trucks on the performance of LNG-DIRP, we consider several instances and run the matheuristic using different fleet sizes. In the instance *NL*_15_7 there are four trucks and four barges. We

Table 13
The effects of varying the number of trucks.

# of trucks	Distance traveled (km)		Cost (€)					Utilization (%)		# of visits	
	Truck	Barge	Total	Purchasing	Routing	Filling stations	Storage facility	Truck	Barge	Truck	Barge
2	4,016	1,716	61,882	14,738	17,018	23,822	6,304	91.8	72.2	32	20
3	4,286	1,512	61,370	14,773	17,520	23,096	5,981	86.7	61.8	35	19
4	6,071	1,357	60,729	14,583	17,493	22,987	5,665	90.9	34.3	47	9
5	7,914	0	60,206	14,679	16,190	23,030	6,307	88.3	-	55	-
6	8,342	0	59,957	14,584	16,915	22,519	5,939	88.3	-	58	-

Table 14
The effects of varying the number of barges.

# of barges	Distance traveled (km)		Cost (€)					Utilization (%)		# of visits	
	Truck	Barge	Total	Purchasing	Routing	Filling stations	Storage facility	Truck	Barge	Truck	Barge
2	6,270	983	60,655	14,721	16,564	23,331	6,039	94.3	45.0	45	8
3	6,773	645	60,816	14,766	16,905	22,942	6,203	86.3	98.2	47	8
4	6,071	1,357	60,729	14,583	17,493	22,987	5,665	90.9	34.3	47	9
5	7,002	641	60,688	14,707	17,206	22,395	6,380	88.9	87.7	48	8
6	5,260	1,550	60,496	14,553	17,565	22,713	5,665	93.4	39.0	45	12

Table 15
The effects of fleet size on the results of the proposed algorithm.

Instance	# of vehicles		Dist. traveled (km)		Cost (€)					Utilization (%)		# of visits	
	Truck	Barge	Truck	Barge	Total	Purchasing	Routing	Filling stations	Storage facility	Truck	Barge	Truck	Barge
NL_10_2	6	0	1,459	0	11,461	2,759	2,984	4,515	1,203	84.8	-	10	0
	4	2	1,459	0	11,461	2,759	2,984	4,515	1,203	84.8	-	10	0
	0	4	0	413	12,000	2,759	3,523	4,515	1,202	-	73.6	0	10
NL_20_14	6	0	29,015	0	171,186	37,428	65,737	59,209	8,812	87.5	-	140	0
	3	3	15,352	1,053	175,969	38,054	52,038	72,686	13,191	90.9	80.5	97	16
	0	5	0	6,788	193,558	38,709	52,612	91,640	10,597	-	57.7	0	84
NL_40_7	10	0	19,456	0	170,238	35,612	55,985	66,046	12,595	90.9	-	152	0
	6	5	17,587	3,270	176,005	35,783	71,319	60,146	8,757	94.0	78.4	101	47
	0	8	0	7,014	198,290	35,393	75,572	77,861	9,464	-	64.2	0	113
NL_50_14	22	0	95,926	0	424,396	96,093	155,365	136,190	36,748	92.2	-	569	0
	22	1	83,555	6,525	412,748	95,930	151,729	148,404	16,685	87.2	89.4	316	70
	17	2	65,155	5,223	413,561	96,048	133,571	165,349	18,593	91.7	84.0	309	67
	15	9	68,550	17,093	421,986	95,874	160,936	148,379	16,797	87.6	83.8	263	125
	0	14	0	13,871	458,427	100,494	118,401	220,938	18,594	-	80.9	0	225

first change the number of trucks in the range from two to six and run the model on these new instances. In the next experiment, we keep four trucks but vary the number of barges from two to six. Tables 13 and 14 present the results of these two experiments. The results show that this specific instance is rather sensitive to the number of trucks. With the increase in the number of trucks, the total cost of the solution improves and traditional trucks replace barges. Routing costs also decline with the increase in the number of trucks. This pattern, however, is not observed when adding to the number of barges. This is due to the fact that the nodes in this instance are more accessible by trucks than barges.

We perform similar analyses on four other instances, however, to avoid a lengthy report, we only illustrate a summary of the analyses. Table 15 shows that, except for *NL*_50_14, the other instances considered for this experiment give better results with a fleet of trucks. The instance *NL*_50_14, however, would need at least one barge for better performance. This is due to a better accessibility of several nodes by barge.

5.7. The effects of deterioration

Deteriorating items include a diverse range of products. While LNG has a fairly low rate of deterioration (evaporation), this is not the case for a large number of deteriorating items. In order to study the impact of deterioration on the solution of the matheuristic, we consider several instances and vary the deterioration rate in a way that could also be applicable to items such as cut flowers, fresh food, and vegetables.

Table 16
The effects of deterioration.

Instance (# of trucks)	Det. rate (%)	Distance traveled (km)	Cost (€)					Utilization (%)	# of visits
			Total	Purchasing	Routing	Filling stations	Storage facility		
NL_10_14 (4)	0	13,691	77,575	19,812	19,612	28,541	9,610	78.3	64
	6	12,676	82,947	21,409	20,209	29,853	11,476	86.2	77
	15	14,047	87,751	23,695	22,368	29,693	11,995	83.7	85
NL_35_7 (8)	0	14,179	140,346	34,648	39,125	58,956	7,617	93.7	117
	6	15,624	145,812	37,721	45,137	55,407	7,547	93.2	140
	15	17,688	152,268	40,324	52,382	50,930	8,632	90.3	166
NL_40_14 (10)	0	40,689	330,977	82,121	96,149	120,030	32,677	90.0	344
	6	43,342	341,298	89,025	104,914	119,320	28,039	91.6	385
	15	45,015	350,325	96,916	117,076	119,562	26,773	91.2	450
NL_50_7 (10)	0	14,606	183,611	51,567	50,693	74,183	7,168	94.0	176
	6	15,953	193,293	53,948	55,267	74,334	9,744	92.1	192
	15	17,831	204,171	57,737	65,785	70,717	9,932	93.3	239

In order to isolate the effects of deterioration, in each instance, we set the number of barges equal to zero and consider a number of trucks that guarantees a feasible solution. Table 16 shows the effects of deterioration on the solution. In this problem, shortages are not allowed, therefore the system aims at a service level of 100%. It is, then, intuitive that a higher deterioration rate would result in a higher purchasing and total cost.

6. Conclusions

We described a matheuristic to solve the deteriorating inventory management problem arising in liquefied natural gas (LNG) distribution. The algorithm uses a mathematical model to generate an initial solution and then applies adaptive large neighborhood search algorithm to improve the solution quality. The enhanced ALNS uses new, as well as existing removal and insertion operators. The break-move operators were especially designed for the inventory aspect of the problem. To fully evaluate the effectiveness of the matheuristic, we generated different sets of instances based on real geographic data of the Netherlands and we compiled a library of instances. We presented results of extensive computational experiments using the proposed algorithm and we compared it against the solutions produced by solving the integer linear programming formulation of the problem by CPLEX. Our results show that the proposed algorithm is highly effective in finding good-quality solutions on instances with up to 100 filling stations and 14 time periods. This research also highlighted the following four important conclusions.

- Apart from cost minimization (CM), we examined two alternative replenishment policies for the inland LNG distribution systems. In the first alternative policy, the transport planner aims at fully utilizing the vehicles' capacity (FVL) while the second alternative policy focuses on delivering maximum quantity (MQD) when visiting a filling station. The results of the study show that since the focus of the policies is on high utilization of the vehicles, the routing costs of the policies are much smaller than those obtained under the original cost minimization policy. The instances studied for this purpose showed an increase of up to 15% in the total cost function when one of the alternative policies is implemented. These policies, intuitively, reduce the number of visits over the planning horizon, which for the case of hazardous item distribution such as LNG could be important for the regulators.
- For each instance presented in Tables 10 and 11, we calculated the percentage of distance traveled by each type of vehicles, e.g. for the instance *NL*_50_7, distances traveled by trucks and barges are 14,308 km and 1,126 km, which result in 92.7% and 7.3% contribution by each mode of transportation. Taking an average over the contribution values by each mode, we observed that the share of barges for medium-size instances is 19%. This share increased to 37% for large-size instances presented in Table 11. This is particularly crucial for companies who would like to make strategic decisions on how much to invest on barge transportation in the context of the growing market share of LNG.
- In the instances used for this research, due to accessibility limitations, barges could be used for only a fraction of the deliveries. The effects of this limitation could be seen when comparing the average utilization of the trucks and barges in medium and large instances. The increase in utilization of the barges in the larger instances (around 15%) could be associated to the high number of cities connected to the waterways in those instances. In the smaller instances, a higher average utilization of barges is not achievable since this results in larger quantities (to cover the demand of filling stations over larger number of periods), and hence in higher holding cost (the storage facility has a lower inventory holding cost).
- Our analyses highlight the impact that deterioration can have on the performance of the LNG distribution network. The effects of evaporation rate on the solution of the LNG-DIRP could be as high as 13% for items with higher rate of deterioration compared to the case with no deterioration. For higher deterioration rates, the model tries to make trade-offs between holding and routing costs, i.e. making more frequent replenishments to the filling stations with smaller quantities delivered to avoid higher deterioration costs.

Acknowledgements

The authors gratefully acknowledge fundings provided by VU University Amsterdam, Dutch Institute for Advanced Logistics (Dinalog), Cardiff University, Eindhoven University of Technology, the Research Council of Norway through the Axiom project, and the Canadian National Sciences and Engineering Research Council under grant 2015-01689. The authors are grateful to anonymous reviewers for their constructive comments and suggestions to improve the content and the presentation of this paper. Also, special thanks are due to Professor Hai Yang for his invaluable advice.

References

Agra, A., Christiansen, M., Hvattum, L.M., Rodrigues, F., 2018. Robust optimization for a maritime inventory routing problem. Transp. Sci. 52 (3), 509–525.
Al-Haidous, S., Msakni, M.K., Haouari, M., 2016. Optimal planning of liquefied natural gas deliveries. Transp. Res. Part C 69, 79–90.
Andersson, H., Christiansen, M., Desaulniers, G., 2016. A new decomposition algorithm for a liquefied natural gas inventory routing problem. Int. J. Prod. Res. 54 (2), 564–578.
Andersson, H., Christiansen, M., Desaulniers, G., Rakke, J.G., 2017. Creating annual delivery programs of liquefied natural gas. Optim. Eng. 18 (1), 299–316.
Bakker, M., Riezebos, J., Teunter, R.H., 2012. Review of inventory systems with deterioration since 2001. Eur. J. Oper. Res. 221, 275–284.
Bertazzi, L., Savelsbergh, M.W.P., Speranza, M.G., 2008. Inventory Routing. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), The vehicle routing problem: Latest advances and new challenges. vol 43. Springer, New York, pp. 49–72.
Chitsaz, M., Divsalar, A., Vansteenwegen, P., 2016. A two-phase algorithm for the cyclic inventory routing problem. Eur. J. Oper. Res. 254 (2), 410–426.
Cho, J., Lim, G.J., Kim, S.J., Biobaku, T., 2018. Liquefied natural gas inventory routing problem under uncertain weather conditions. Int. J. Prod. Econ. 204, 18–29.
Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2013. Ship routing and scheduling in the new millennium. Eur. J. Oper. Res. 228, 467–483.
Coelho, L.C., Cordeau, J.–.F., Laporte, G., 2014. Thirty years of inventory routing. Transp. Sci. 48 (1), 1–19.
Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem.. Eur. J. Oper. Res. 223 (2), 346–359.
DINALOG, 2018. Design of LNG networks. https://www.dinalog.nl/project/design_of_lng_networks-1 (Accessed: 2019-02-01).
Dobrota, Đ., Lalić, B., Komar, I., 2013. Problem of boil-off in LNG supply chain. Trans.Maritime Sci. 2 (02), 91–100.
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. J. Energy Markets 3 (4), 31–64.
Foss, M.M., Delano, F., Gulen, G., Makaryan, R., 2003. LNG Safety and Security. Center for Energy Economics (CEE).
Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., Stobbe, M., 2017. A metaheuristic for the time-dependent pollution-routing problem. Eur. J. Oper. Res. 259 (3), 972–991.
Gedde, S.T., 2014. Contractual and Economical Consequences of LNG Boil-Off Quality. Department of Marine Technology, Norwegian University of Science and Technlogy, Trondheim, Norway Master's thesis.
Gerdsmeyer, K., Isalski, W., 2005. On-board reliquefaction for lng ships. In: Proc. of the Gas Processors Association Europe Conference, London, May 2005.
Ghiami, Y., Beullens, P., 2016. Planning for shortages? net present value analysis for a deteriorating item with partial backlogging. Int. J. Prod. Econ. 178, 1–11.
Ghiami, Y., Van Woensel, T., Christiansen, M., Laporte, G., 2015. A Combined Liquefied Natural Gas Routing and Deteriorating Inventory Management Problem. In: Corman, F., Voß, S., Negenborn, R. (Eds.), Computational Logistics. ICCL 2015. Lecture Notes in Computer Science. vol 9335. Springer, Cham, pp. 91–104.
Ghilas, V., Demir, E., Van Woensel, T., 2016. A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. Transp. Res. Part B 91, 34–51.
Goel, V., Furman, K.C., Song, J.-H., El-Bakry, A.S., 2012. Large neighborhood search for lng inventory routing. J. Heuristics 18 (6), 821–848.
Grønhaug, R., Christiansen, M., Desaulniers, G., Desrosiers, J., 2010. A branch-and-price method for a liquefied natural gas inventory routing problem. Transp. Sci. 44 (3), 400–415.
Halvorsen-Weare, E.E., Fagerholt, K., 2013. Routing and scheduling in a liquefied natural gas shipping problem with inventory and berth constraints. Ann. Oper. Res. 203 (1), 167–186.
Halvorsen-Weare, E.E., Fagerholt, K., Rönnqvist, M., 2013. Vessel routing and scheduling under uncertainty in the liquefied natural gas business. Comput. Ind. Eng. 64, 290–301.
Hemmati, A., Hvattum, L.M., Christiansen, M., Laporte, G., 2016. An iterative two-phase hybrid matheuristic for a multi-product short sea inventory-routing problem. Eur. J. Oper. Res. 252 (3), 775–788.
Janssen, L., Claus, T., Sauer, J., 2016. Literature review of deteriorating inventory models by key topics from 2012 to 2015. Int. J. Prod. Econ. 182, 86–112.
Jones, J., 1974. Reliquefaction of boil off gas. US Patent 3,857,245
Kim, D., Hwang, C., Gundersen, T., Lim, Y., 2019. Process design and economic optimization of boil-off-gas re-liquefaction systems for LNG carriers. Energy 173, 1119–1129.
Moon, J., Lee, Y., Jin, Y., Hong, E., Chang, H., 2007. Cryogenic refrigeration cycle for re-liquefaction of LNG boil-off gas. In: International cryocooler conference, pp. 629–635.
Mutlu, F., Msakni, M.K., Yildiz, H., Sönmez, E., Pokharel, S., 2016. A comprehensive annual delivery program for upstream liquefied natural gas supply chain. Eur. J. Oper. Res. 250 (1), 120–130.
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. Eur. J. Oper. Res. 234 (2), 350–366.
Planas, E., Pastor, E., Casal, J., Bonilla, J., 2015. Analysis of the boiling liquid expanding vapor explosion (bleve) of a liquefied natural gas road tanker: the zarzalico accident. J. Loss Prev. Process Ind. 34, 127–138.
Rakke, J.G., Andersson, H., Christiansen, M., Desaulniers, G., 2015. A new formulation based on customer delivery patterns for a maritime inventory routing problem. Transp. Sci. 49 (2), 384–401.
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. Transp. Res. Part C 19, 896–911.
Rohmer, S., Claassen, G., Laporte, G., 2019. A two-echelon inventory-routing problem for perishable products. Comput. Oper. Res..
Roldán, R.F., Basagoiti, R., Coelho, L.C., 2016. Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. Comput. Oper. Res. 74, 14–20.
Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.
Shao, Y., Furman, K.C., Goel, V., Hoda, S., 2015. A hybrid heuristic strategy for liquefied natural gas inventory routing. Transp. Res. Part C 53, 151–171.
Shell, 2018. Shell LNG Outlook. Technical Report.
Shin, M.W., Shin, D., Choi, S.H., Yoon, E.S., 2008. Optimal operation of the boil-off gas compression process using a boil-off rate model for lng storage tanks. Korean J. Chem. Eng. 25 (1), 7–12.
Soysal, M., Bloemhof-Ruwaard, J.M., Haijema, R., van der Vorst, J.G., 2018. Modeling a green inventory routing problem for perishable products with horizontal collaboration. Comput. Oper. Res. 89, 168–182.

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. Comput. Ind. Eng. 62 (1), 245–255.
SteadieSeifi, M., Dellaert, N.P., Nuijten, W., Van Woensel, T., 2017. A metaheuristic for the multimodal network flow problem with product quality preservation and empty repositioning. Transp. Res. Part B 106, 321–344.
Uggen, K.T., Fodstad, M., Nørstebø, V.S., 2013. Using and extending fix-and-relax to solve maritime inventory routing problems. TOP 21 (2), 355–377.
Zhang, Z.-H., Unnikrishnan, A., 2016. A coordinated location-inventory problem in closed-loop supply chain. Transp. Res. Part B 89, 127–148.