Optimization Letters (2022) 16:211–235
https://doi.org/10.1007/s11590-020-01630-y

ORIGINAL PAPER

Variable neighborhood search-based solution methods for the pollution location-inventory-routing problem

Panagiotis Karakostas[1] · Angelo Sifaleras[2] · Michael C. Georgiadis[1]

Received: 23 March 2020 / Accepted: 5 August 2020 / Published online: 26 August 2020

Abstract

This work presents efficient solution approaches for a new complex NP-hard combinatorial optimization problem, the Pollution Location Inventory Routing problem (PLIRP), which considers both economic and environmental issues. A mixed-integer linear programming model is proposed and first, small problem instances are solved using the CPLEX solver. Due to its computational complexity, General Variable Neighborhood Search-based metaheuristic algorithms are developed for the solution of medium and large instances. The proposed approaches are tested on 30 new randomly generated PLIRP instances. Parameter estimation has been performed for determining the most suitable perturbation strength. An extended numerical analysis illustrates the effectiveness and efficiency of the underlying methods, leading to high-quality solutions with limited computational effort. Furthermore, the impact of holding cost variations to the total cost is studied.

Keywords Fuel consumption · Location · Inventory · Routing · Variable neighborhood search

✉ Michael C. Georgiadis
mgeorg@auth.gr

Panagiotis Karakostas
pkarakost@cheng.auth.gr

Angelo Sifaleras
sifalera@uom.gr

[1] Department of Chemical Engineering, Aristotle University of Thessaloniki, University Campus, 54124 Thessaloniki, Greece

[2] Department of Applied Informatics, School of Information Sciences, University of Macedonia, 156 Egnatia Str., 54636 Thessaloniki, Greece

1 Introduction

The Location Inventory Routing Problem (LIRP) is a complex NP-hard combinatorial optimization problem, which simultaneously tackles strategic (location/allocation), tactical (inventory levels and replenishment rates) and operational (routing schedules) decisions [20,30,39]. The main goal of this problem is to determine an optimal schedule for achieving economic benefits, like total cost minimization [18]. However, due to the fact that the supply chain activities emit pollutants, such as carbon dioxide (CO_2), the environmental impact of logistics should also be taken under consideration [7,25]. More specifically, CO_2 emissions are considered as the main cause of the global warming, one of the major environment challenges [7,8]

Freight transportation is mentioned as the main source of CO_2 [27]. Especially, road transportation generates more than 20% of the total CO_2 emissions in European Union [27] and 30% in the Organization for Economic Co-operation and Development (OECD) countries [31]. CO_2 emissions are proportional to the amount of consumed fuel and the fuel consumption depends on speed, acceleration, distance and total weight of the vehicle (curb and freight weight) [25].

The green logistics concept has been studied in many literature contributions. The majority of them focus on routing problems, such as the Vehicle Routing Problem [4,25,29,32]. However, the structure of a supply chain network can affect its functional efficiency by leading to unnecessary delivery routes or redundant facilities [1]. A limited amount of works study the effect of environmental impact consideration on the design of logistics optimization problems [11]. Koc et al. [26] investigated the combined impact of depot location, fleet composition and routing decisions on vehicle emissions especially in last mile deliveries. They developed an Adaptive Large Neighborhood Search (ALNS) metaheuristic and applied it on several instances with up to 100 customers. They, concluded that circuitous routes can lead to faster speeds and lower costs and CO_2 emissions. Toro et al. [36] proposed a bi-objective mixed-integer-linear programming (MILP) model for the Green Capacitated Location Routing Problem (G-CLRP). They observed that the use of more vehicles leads to significant fuel economy in the long term. Cheng et al. [6] examined the impact of four carbon emission regulation policies. They proposed mixed-integer-nonlinear programming (MINLP) models and linearization methods along with a Genetic Algorithm (GA) for solving various problem instances. Furthermore, several managerial insights were reported by extensive sensitivity analyses. Cheng et al. [7] introduced a Green Inventory Routing Problem (GIRP) with heterogeneous fleet of vehicles in which environmental impacts were taken into consideration. An exact branch-and-cut algorithm was developed for solving various problem instances and reporting managerial insights.

Despite the significance of considering the environmental impact of logistic activities there is a lack of works on more complex and potentially realistic problems, such as the Location Inventory Routing Problem (LIRP). Zhalechian et al. [38] proposed a multi-objective MINLP model for the closed-loop LIRP. A stochastic–possibilistic approach was used for handling the problem uncertainty. A hybrid self-adaptive GA and a Variable Neighborhood Search (VNS) metaheuristic algorithm was developed in order to solve large-sized instances. Karakostas et al. [22] introduced a Pollution

LIRP and proposed a Basic VNS (BVNS) metaheuristic algorithm for the solution of medium-sized problem instances.

This work presents a Pollution LIRP, which considers both economic and environmental impacts of the main logistic activities, such as facilities location, inventory control and vehicle routing. The main contributions of this work are summarized as follows:

- A new complex logistics optimization problem with environmental considerations is proposed.
- An MIP model is formulated by integrating and modifying two models from the literature [7,39].
- Driving wages are taken under consideration.
- Three VNS-based algorithms have been developed.
- A new benchmark set with the current largest instances of the Pollution LIRP (PLIRP) reported in the open literature have been generated and made publicly available.
- The impact of the flexible replenishment policy on building better routing patterns is illustrated.
- A sensitivity analysis is performed for testing how the variation on holding costs can affect the objective values of the problem instances.

The current work is structured as follows. Sections 2 and 3 present the problem statement and the proposed solution algorithm, respectively. Section 4 provides the computational analysis for evaluating the performance of the proposed method. Finally, Sect. 5 draws up concluding remarks and some thoughts on potential future extensions.

2 Problem description

The PLIRP is defined as a two-echelon supply chain network. Given:

- a set of time periods,
- a set of potential capacitated depots,
- a set of geographically distributed customers,
- a set of homogeneous capacitated fleet of vehicles,
- a single type of product,
- a period-variable demand of each customer

Determine:

- the number and location of depots to be established,
- the allocation of customers to the opened depots,
- the inventory levels at each customer,
- the replenishment quantities and rates for each customer,
- the routes of vehicles,
- the selection of a speed level for traveling each link of the scheduled network.

In order to: minimize an objective function representing the total cost.

The key model assumption are as follows:

Table 1 Sets of the mathematical model

Indices	Explanation
V	Set of nodes
J	Set of candidate depots
I	Set of customers
K	Set of vehicles
H	Set of discrete and finite planning horizon
R	Set of speed levels

- each customer is serviced by one depot,
- each customer is serviced by at most one vehicle in each time period,
- a vehicle departs from and returns to the same depot after servicing one or more customer(s),
- the total delivered quantity to each customer over the time horizon must be equal to its total demand,

The problem is formulated as a mixed integer programming model, by integrating an LIRP model [39] with a Green Inventory-Routing problem (G-IRP) model [7]. Its sets, parameters and variables are provided in Tables 1, 2, 3, and 4.

The value of the parameter f_c is calculated as the average of the petrol prices in 40 European countries taken from the website www.globalpetrolprices.com in 26th of February in 2018. The value of parameters f_e and f_d are converted into Euro currency (26th of February, 2018).

The objective of the problem is the minimization of total supply chain system cost, including the following cost components:

- Location Cost: $\sum_{j\in J} f_j y_j$, which represents the cost of opening the needed number of depots.
- Inventory Cost $\sum_{i\in I} h_i \sum_{t\in H}\left(\frac{1}{2}d_{it} + \sum_{p\in H,p<t} w_{itp}\,(t-p) + \sum_{p\in H,p>t} w_{itp}\,(t-p+|H|)\right)$. It consists of three cost components. The first component represents the average inventory holding cost. The remaining terms impose penalty costs for any early or late replenishment.
- Routing Cost: $\sum_{i\in V}\sum_{j\in V}\sum_{t\in H}\sum_{k\in K} c_{ij}x_{ijkt}$. It represents general routing costs, such as vehicles' maintenance and/or insurance costs.
- Fuel Consumption Cost: $\sum_{i\in V}\sum_{j\in V}\sum_{k\in K}\sum_{t\in H}\left\{\lambda\,(f_c+(f_e\sigma))\left(\sum_{r\in R}\frac{(zz_{ijktr}\ EFF_k\ ES_k\ ED_k\ c_{ij})}{s_r}+\left(\alpha\gamma_k\left(CW_k\ x_{ijkt}+a_{ijkt}\right)c_{ij}\right)+\left(\beta_k\ \gamma_k\sum_{r\in R}\left(s_r\ zz_{ijktr}\right)^2\right)\right)\right\}$. The Comprehensive Modal Emission Model (CMEM) is adopted [3]. Thus, the fuel consumption is affected by vehicle specific characteristics, such as the weight of the load, the vehicle's speed and obviously the traveling distance. More specifically, the following formulas are utilized.
  - $\lambda = \frac{\epsilon}{HVDF*\psi}$
  - $\gamma_k = \frac{1}{1000\ VDTE\ \eta}$

Table 2 Vehicles' parameters

Parameter	Explanation	Value
ϵ	Fuel-to-air mass ratio	1[a]
g	Gravitational constant (m/s^b)	9.81[a]
ρ	Air density (kg/m^3)	1.2041[a]
CR	Coefficient of rolling resistance	0.01[a]
η	Efficiency parameter for diesel engines	0.45[a]
f_c	Unit fuel cost (L)	1.3
f_e	Unit CO_2 emission cost (kg)	0.2793[a]
f_d	Driver wage (s)	0.0025[a]
σ	CO_2 emitted by unit fuel consumption (kg/L)	2.669[a]
$HVDF$	Heating value of a typical diesel fuel (kj/g)	44[a]
ψ	Conversion factor (g/s to L/s)	737[a]
θ	Road angle	0[a]
τ	Acceleration (m/s^2)	0[a]
CW_k	Curb weight (kg)	3500[b]
EFF_k	Engine friction factor (kj/rev/L)	0.25[a]
ES_k	Engine speed (rev/s)	39[a]
ED_k	Engine displacement (L)	2.77[a]
CAD_k	Coefficient of aerodynamics drag	0.6[a]
FSA_k	Frontal surface area (m^2)	9[a]
$VDTE_k$	Vehicle drive train efficiency	0.4[a]

[a] [7]
[b] [25]

Table 3 Rest PLIRP model parameters

Notation	Explanation
f_j	Fixed opening cost of depot j
C_j	Storage capacity of depot j
h_i	Unit inventory holding cost of customer i
Q_k	Loading capacity of vehicle k
d_{it}	Demand of customer i in period t
c_{ij}	Distance of locations pair (i, j)
s_r	The value of the speed level r
PW	Weight of each product unit (1.5 kg)
M	Classic big-M parameter (100,000)

- $\alpha = \tau + g\ CR\ \sin\theta + g\ CR\ \cos\theta$
- $\beta_k = 0.5\ CAD\ \rho\ FSA_k$

The first component is the fuel consumption based on the vehicles' engine function, while the second cost term represents the cost of consumed fuel because of the

Table 4 PLIRP model variables

Notation	Explanation
y_j	1 if j is opened; 0 otherwise
z_{ij}	1 if customer i is assigned to depot j; 0 otherwise
x_{ijkt}	1 if node j is visited after i in period t by vehicle k
q_{ikt}	Product quantity delivered to customer i in period t by vehicle k
w_{itp}	Quantity delivered to customer i in period p to satisfy its demand in period t
a_{vikt}	Load weight by travelling from node v to the customer i with vehicle k in period t
$zz_{v_1 v_2 ktr}$	1 if vehicle k travels from node $v1$ to $v2$ in period t with speed level r

total vehicles' weight (curb weights plus load weights). The third one represents the fuel consumption cost related to the vehicles' speed levels.

- Driver Wages Cost: $\sum_{i\in V}\sum_{j\in V}\sum_{k\in K}\sum_{t\in H}\sum_{r\in R} f_d \frac{(zz_{ijktr}\ c_{ij})}{s_r}$, which represents the cost of the drivers wages.

Thus, the total cost (TC) is calculated as: TC = Location Cost + inventory Cost + Routing Cost + Fuel Consumption Cost + Driver Wages Cost.

The mathematical formulation, of the problem under consideration, is as follows:

$$\min \quad TC \tag{1}$$

Subject to

$$\sum_{r\in R} zz_{ijktr} = 1 \quad \forall i, j \in V, \forall k \in K, \quad \forall t \in H \tag{2}$$

$$\sum_{i\in V} a_{ijkt} - \sum_{i\in V} a_{jikt} = q_{jkt} PW \quad \forall j \in I, \forall k \in K, q \quad \forall t \in H \tag{3}$$

$$\sum_{j\in V} x_{ijkt} - \sum_{j\in V} x_{jikt} = 0 \quad \forall i \in V, \forall k \in K, \forall t \in H \tag{4}$$

$$\sum_{j\in V}\sum_{k\in K} x_{ijkt} \leq 1 \quad \forall t \in H, \quad \forall i \in I \tag{5}$$

$$\sum_{j\in V}\sum_{k\in K} x_{jikt} \leq 1 \quad \forall t \in H, \quad \forall i \in I \tag{6}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijkt} \quad \leq 1 \quad \forall k \in K, \quad \forall t \in H \tag{7}$$

$$x_{ijkt} = 0 \quad \forall i, j \in J, \quad \forall k \in K, \quad \forall t \in H, \quad i \neq j \tag{8}$$

$$\sum_{i\in I} q_{ikt} \leq Q_k \quad \forall k \in K, \quad \forall t \in H \tag{9}$$

$$\sum_{j\in J} z_{ij} = 1 \quad \forall i \in I \tag{10}$$

$$z_{ij} \leq y_j \quad \forall i \in I, \quad \forall j \in J \tag{11}$$

$$\sum_{i\in I}\left(z_{ij}\sum_{t\in H}d_{it}\right)\leq C_j \quad \forall j\in J \tag{12}$$

$$\sum_{u\in I}x_{ujkt}+\sum_{u\in V\backslash\{i\}}x_{iukt}\leq 1+z_{ij} \quad \forall i\in I,\quad \forall j\in J,\quad \forall \mathrm{k}\in \mathrm{K},\ \forall t\in H \tag{13}$$

$$\sum_{i\in I}\sum_{k\in K}\sum_{t\in H}x_{jikt}\geq y_j \quad \forall j\in J \tag{14}$$

$$\sum_{i\in I}x_{jikt}\leq y_j \quad \forall j\in J,\forall k\in K,\forall t\in H \tag{15}$$

$$\sum_{p\in H}w_{itp}=d_{it} \quad \forall i\in I,\quad \forall \mathrm{t}\in \mathrm{H} \tag{16}$$

$$\sum_{t\in H}w_{itp}=\sum_{k\in K}q_{ikp} \quad \forall i\in I,\quad \forall p\in H \tag{17}$$

$$q_{ikt}\leq M\sum_{j\in V}x_{ijkt} \quad \forall i\in I,\quad \forall t\in H,\quad \forall k\in K \tag{18}$$

$$\sum_{j\in V}x_{ijkt}\leq Mq_{ikt} \quad \forall i\in I,\quad \forall t\in H,\quad \forall \mathrm{k}\in \mathrm{K} \tag{19}$$

$$\mathrm{x}_{ijkt}\in\{0,1\} \quad \forall i\in I,\forall j\in J,\quad \forall t\in H,\quad \forall k\in K \tag{20}$$

$$y_j\in\{0,1\} \quad \forall j\in J \tag{21}$$

$$z_{ij}\in\{0,1\} \quad \forall i\in I,\quad \forall j\in J \tag{22}$$

$$q_{ikt}\leq \min\left\{Q_k,\sum_{p\in H}d_{ip}\right\} \quad \forall i\in I,\quad \forall j\in J,\forall k\in K \tag{23}$$

$$w_{itp}\leq d_{ip} \quad \forall i\in I,\quad \forall t,p\in H \tag{24}$$

Constraints (2) impose that a vehicle travels between nodes with a specific speed level in each time period. Constraints (3) act as subtour elimination constraints, as they declare that the difference between the total weight of the incoming flow of product to a selected customer and the total weight of the outcoming product flow of that customer equals the product weight delivered to that customer in the selected time period with the selected vehicle. The equilibrium between the interior and exterior flow of vehicles is guaranteed by Constraints (4). Constraints (5) and (6) ensure that exactly one vehicle visits each customer at each time period. Constraints (7) guarantee that a vehicle performs mostly one route at each time period. Constraints (8) forbid the movement of a vehicle between depots. Constraints (9) ensure that, the total amount of products sent by a vehicle at a specific period does not exceed the capacity of that vehicle. Constraints (10) guarantee that a vehicle will be travelled from a depot to a customer only if that customer is allocated to the depot. Constraints (11) impose that a customer is assigned to a depot only if that depot is selected to be opened. Constraints (12) respect the capacity of depots. A customer is connected to a depot, only if that customer is assigned to that depot, according to Constraints (13). A vehicle departures from a depot only if that depot is opened according to Constraints (14) and (15).

The delivered amount of product to each customer at each time period satisfies the demand of that customer, as guaranteed by Constraints (16). Constraints (17) ensure the equilibrium between scheduled and actual deliveries. A customer is visited at a specific period, only if a replenishment is scheduled for that period, according to Constraints (18). The rest constraints of the model declare the nature of the decision variables.

3 Solution approach

3.1 Construction heuristic

The scope of the constructive phase is the generation of a feasible initial solution. In this work, a three-phased construction heuristic has been developed. In the first phase, location and allocation decisions are made. Inventory-routing decisions are determined in the second phase, while the speed levels for traveling through the network nodes are selected in the last phase.

Initially, for taking the location and allocation decisions, a ratio-based selection criterion is applied for opening the required depots while a nearest customer allocation strategy has been employed for the assignment step. For each one of the candidate depots, the ratio $\frac{fixed\_opening\_cost}{Capacity}$ is computed and the depot with the minimum ratio is chosen. If two or more depots have the same ratio, one of them is selected arbitrarily (commonly the first found). According to the customers' allocation process, the nearest, to the opened depot, customer is chosen and in the case that its total demand does not violate the remaining capacity of the depot, the customer is assigned to that depot. This initial step of construction heuristic is completed when all customers have been allocated to the opened depots.

In each time period the allocated, to the opened depots, customers are assigned to vehicles, by considering both their demand and the capacity of vehicles. Then, the route of each utilized vehicle is built in a selected time period by applying the Random Insertion method [13]. According to the inventory decisions, the delivered quantities are set equal to the corresponding demand for each customer at each time period. Obviously, in this initial phase if a customer does not require any quantity of the product in a selected period, he will not be included in any route over that period. In the last phase, the selection of the speed levels for traveling through the nodes of the structured network is randomly performed.

3.2 Neighborhood structures

In this work, five local search operators are used in the improvement phase of each proposed solution method. These neighborhood structures are the following:

Inter-route relocate In this neighborhood structure a selected customer is removed from its route and moved in the next position of an other selected customer, who is assigned to a different route. Those two customers can be allocated to the same or different depots. Both of the selected customers must be visited by vehicles in the

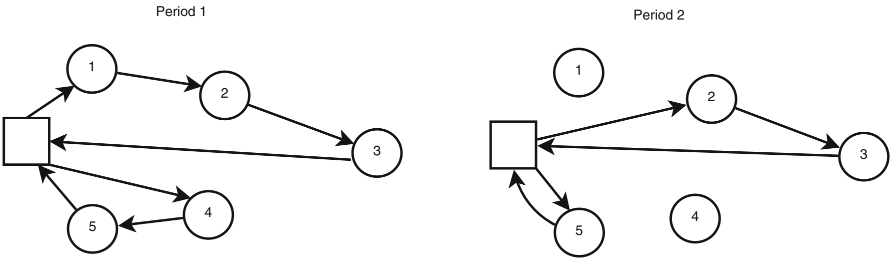

Fig. 1 Routes from the same depot in each time period before the application of the inter-route relocate move

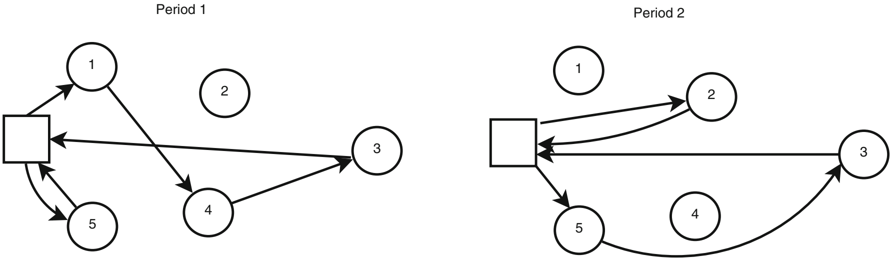

Fig. 2 Routes from the same depot in each time period after the application of the inter-route relocate move

same time periods, in order this move to be applicable. A replenishment shifting may be applied if a vehicle capacity violation occurs. The following cases can be met by applying this neighborhood:

- Case 1: The two selected customers are allocated to the same depot and no vehicle capacity violations occur.
- Case 2: The selected customers are assigned to the same depot and vehicles capacity violations occur.
- Case 3: The two selected customers are assigned to different depots and no vehicle capacity violations occur.
- Case 4: The two selected customers are allocated to different depots and vehicles capacity violations occur.

In the first case only routing decisions are made. In the second case, both routing and inventory decisions are taken, while routing and allocation decisions are considered in the third case of this move. In the last case, routing, inventory and allocation decisions are simultaneously made. Figures 1 and 2illustrate an example of this neighborhood, applied on routes allocated to the same depot. The pairs of customers are (4, 1) in the first time period and (3, 5) in the second time period.

Figures 3 and 4 illustrate an example of this neighborhood, applied on routes allocated to different depots. The move is applied on the pair of customers (2, 5) in all available time periods.

Inter-route exchange This neighborhood swaps two customers from different routes in the time horizon. The exchanged customers can be allocated either to the same depot

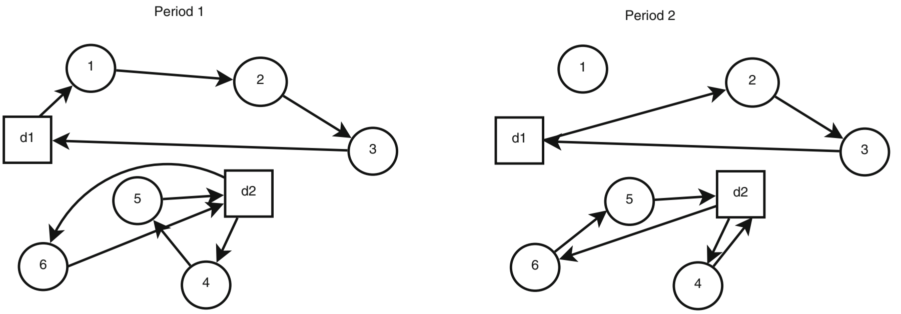

Fig. 3 Routes from different depots in each time period before the application of the inter-route relocate move

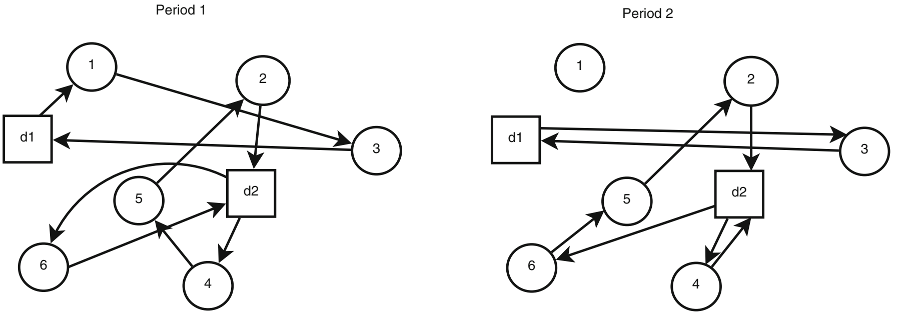

Fig. 4 Routes from different depots in each time period after the application of the inter-route relocate move

or different depots. If the customers are allocated to the same depot, the swapping may not be applied in each time period. However, in the second case the exchanging must be valid for all time periods, in order to be applicable. The special cases of this move are summarized as follows:

- Case 1: Vehicle capacity violation does not occur.
- Case 2: The demand of a customer exceeds the capacity of the vehicle servicing the other customer in one or more time periods.

The first case makes only routing decisions, while in the second case routing and inventory decisions are taken. Changes on allocation decisions will take place only if the swapped customers are allocated to different depots. Figures 5 and 6 present an illustrative example of the inter-route exchange move applied on routes allocated to the same depot for the case of customers (1, 5) in the first period and (2, 4) in the second one.

Figures 7 and 8 depict an instance of the inter-route exchange move applied on routes allocated to different depots for the pair of customers (2, 3).

Exchange opened–closed depots This neighborhood exchanges a closed depot with an opened depot. For a selected closed depot, the cost change for swapping it with each one of all opened depots, is calculated. Then, the pair of the opened–closed depots with the minimum exchanging cost is selected. Subsequently, it is examined if the scheduled swapping does not violate any capacity constraint. After the validation checking step,

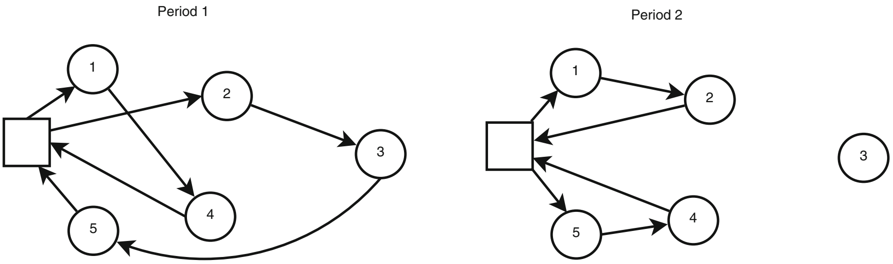

Fig. 5 Routes from the same depot in each time period before the application of the inter-route exchange move

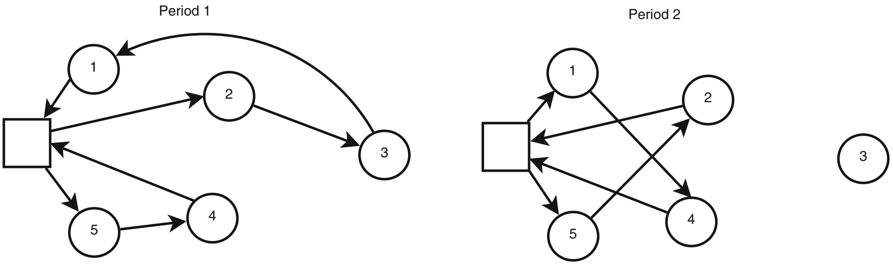

Fig. 6 Routes from the same depot in each time period after the application of the inter-route exchange move

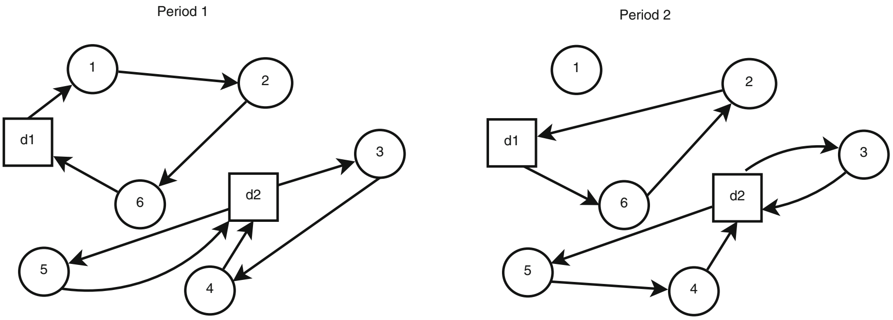

Fig. 7 Routes from different depots in each time period after the application of the inter-route exchange move

the newly opened depot is inserted in the routes assigned to the currently opened depot via a minimum insertion cost procedure. This insertion process may involve customers re-ordering in the routes. Obviously, the move is applied only if the overall cost decreases, which consists of location and routing costs. Figure 9 illustrates the an example of this neighborhood.

Intra-route relocate In this move a selected customer is removed from its current position in its route and moved in a different position in the same route. Figure 10 illustrates an example of the Intra-route Relocate neighborhood, applied on the pair of customers (5, 4).

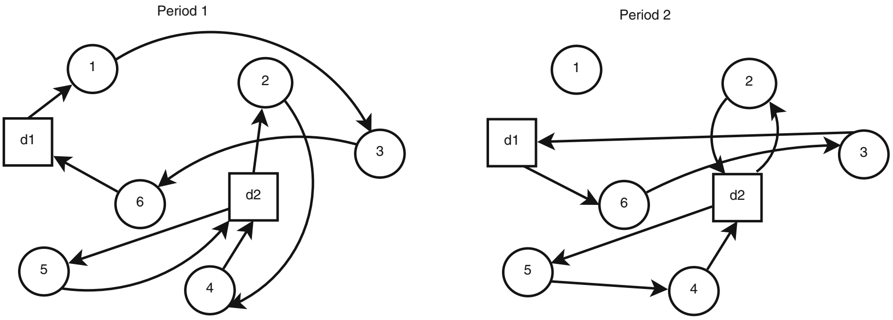

Fig. 8 Routes from different depots in each time period after the application of the inter-route exchange move

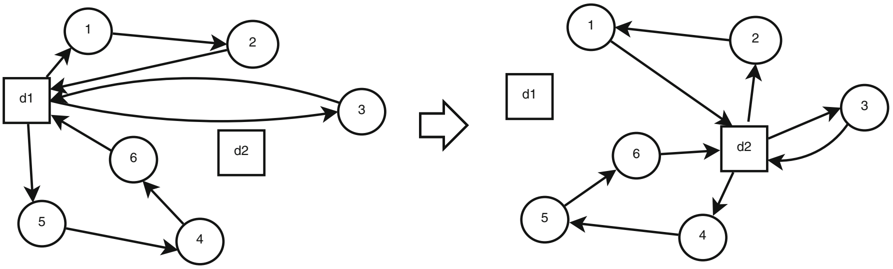

Fig. 9 Example of the exchange opened–closed depots neighborhood

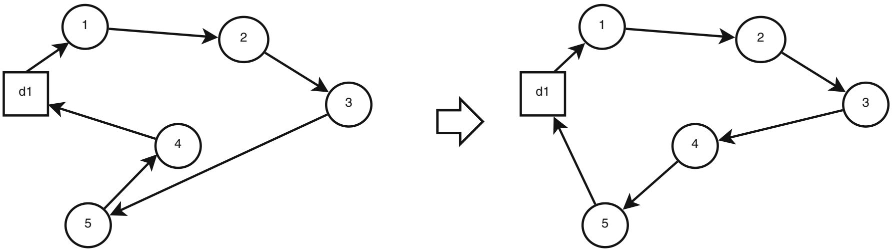

Fig. 10 Example of the intra-route relocate neighborhood

2–2 replenishment exchange In this neighborhood structure, two time periods t_1 and t_2 are randomly selected and the two most distant customers i and b, both serviced in those two periods, are identified. The replenishment of customer i in period t_1 is moved to the period t_2, and the replenishment of customer b is moved from period t_2 to period t_1. Consequently, there is no need to visit customers i and b in periods t_1 and t_2 respectively. If the total cost decreases and there are no violations on the vehicles capacities, the move is applied. To clarify the function of this neighborhood, an illustrative example is provided in Figs. 11 and 12. The most distant customers are 3 and 7. The customer 3 is removed from its route in period 1 and his delivery is

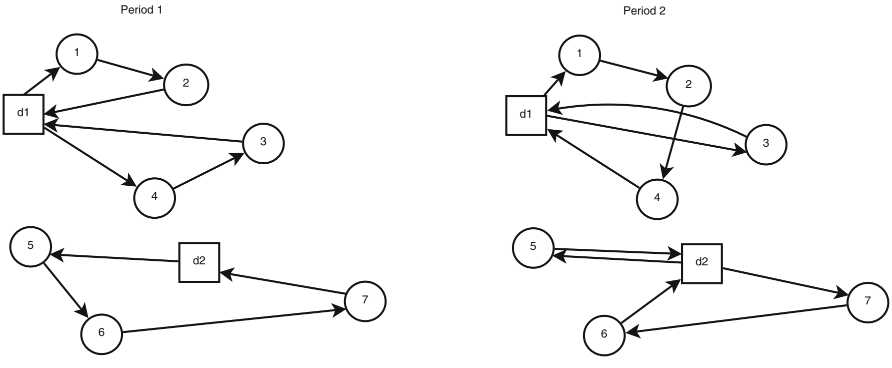

Fig. 11 Example of network structure before the application of 2–2 Replenishment Exchange neighborhood

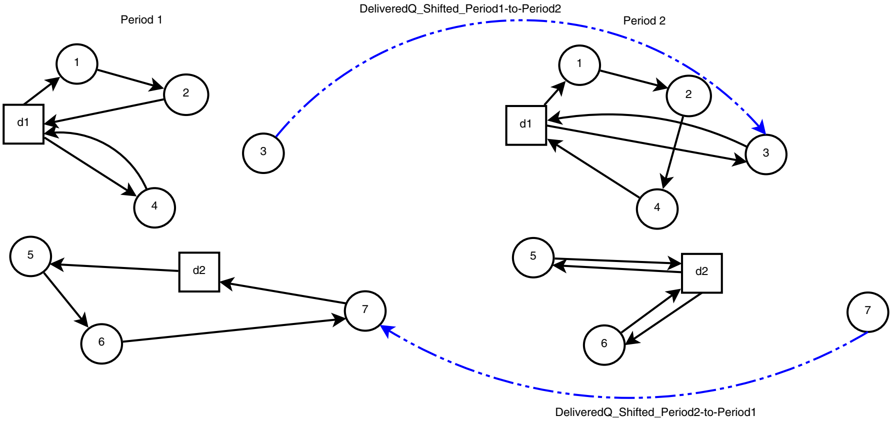
Period 1
DeliveredQ_Shifted_Period1-to-Period2
Period 2
d1
d2
1
2
3
4
5
6
7
DeliveredQ_Shifted_Period2-to-Period1

Fig. 12 Example of network structure after the application of 2–2 Replenishment Exchange neighborhood

shifted in period 2, while the customer 7 is removed from his current route in period 2 and his delivered quantity is shifted in period 1.

In order to avoid potential violation of vehicles capacities, while applying the Inter-route Relocate and the Inter-route Exchange moves, a shifting of surplus product quantity may be applied.

3.3 Shaking procedure

For escaping local optimum solutions, a shaking procedure with three local search operators is proposed, including the following structures:

- Inter-route Exchange.
- Exchange Opened–Closed Depots.
- Intra-Route Relocate.

In each iteration of this diversification method, one of the proposed local search operators is randomly selected and a predefined number of random jumps are applied. Its pseudo-code is given in Algorithm 1.

Algorithm 1 Shaking Procedure

```
procedure SHAKE($S, k, l_{\max}$)
  $l = random\_integer(1, l_{\max})$
  for $i \leftarrow 1, k$ do
    select case($l$)
    case(1)
    $S' \leftarrow Inter-route\_Exchange(S)$
    case(2)
    $S' \leftarrow Exchange\_OpenedClosed\_Depots(S)$
    case(3)
    $S' \leftarrow Intra\_Relocate(S)$
    end select
  end for
  Return $S'$
end procedure
```

The shaking procedure receives an incumbent solution S, the maximum number of iterations $k_{\max}$ executed in the perturbation phase and the number of neighborhood structures $l_{\max}$ as input. A new solution S' is obtained by applying k (where $1 < k < k_{\max}$) times one randomly selected neighborhood of the above local search operators.

3.4 Variable neighborhood search (VNS)

Variable Neighborhood Search (VNS) is a flexible framework for building heuristics, which manages a single solution at each step time [15]. It systematically changes predefined neighborhood structures during the search [9,14,33] and has been successfully applied several times in location, inventory, and routing problems [19,21,24,28].

The Basic VNS (BVNS), the Variable Neighborhood Descent (VND), and the General Variable Neighborhood Search (GVNS) are three of the most well-known variants of the VNS. The BVNS alternates a shaking procedure with a local search operator, while the VND explores several local search operators without using a diversification procedure [15,16,16,34]. According to the neighborhood change criteria, different sequential VND schemes have been proposed. Two of the mostly used VND schemes are the pipe VND (pVND) and the cyclic VND (cVND) [10]. In the pVND the search continues in the same neighborhood, while improvements occurred. In the cVND, the neighborhood structures are alternated regardless the achieved improvements. The GVNS extents the BVNS, by using a VND scheme as its main improvement step [12,23]. In this work, a BVNS, two GVNS (GVNS with cVND and GVNS with pVND) solution methods and their corresponding adaptive variants, have been developed. The execution of the developed VNS-based methods is bounded by a time stopping criterion denoted by the parameter max_time. The proposed solution algorithms are provided in the following pseudo-codes.

Algorithm 2 Basic VNS

```
procedure BVNS($k_{\max}$, $max\_time$, $l_{\max}$)
    $S \leftarrow$ Construction_Heuristic
    $l \leftarrow 1$
    while $time \leq max\_time$ do
        for each neighborhood structure $l$ do
            for $k \leftarrow 1, k_{\max}$ do
                $S' \leftarrow$ Shake($S, k, l_{\max}$)
                $S'' \leftarrow$ Local_Search($S', l$)
                if $f(S'') < f(S)$ then
                    $S \leftarrow S''$
                end if
                $l \leftarrow l + 1$
                if $l > l_{\max}$ then
                    $l \leftarrow 1$
                end if
            end for
        end for
    end while
    return $S$
end procedure
```

Algorithm 3 GVNS$_{\text{pVND}}$

```
procedure GVNS($S, k_{\max}, max\_time, l_{\max}$)
    $S \leftarrow$ Construction_Heuristic
    while $time \leq max\_time$ do
        for $k \leftarrow 1, k_{\max}$ do
            $S* =$ Shake($S, k, l_{\max}$)
            $S' =$ pVND($S*$)
            if $f(S') < f(S)$ then
                $S \leftarrow S'$
            end if
        end for
    end while
    return S
end procedure
```

The pseudo-code of the $GVNS_{cVND}$ algorithm is exactly the same with the pseudo-code of the $GVNS_{pVND}$ algorithm with the only difference that it uses the cVND instead of pVND. The adaptive variants of these methods uses an adaptive re-ordering mechanism of the local search operators. This adaptive mechanism uses past experience, such the number of improvements achieved by each operator and proceeds a different order. More specifically, the array "Improvements_Counter" stores in its positions the improvements achieved by each operator and then a descending sorting is performed on this array by the use of the function $Descending\_Order()$. The pseudo-code of this mechanism is summarized in Algorithm 4.

Algorithm 4 Adaptive_Order

```
procedure ADAPTIVE_ORDER($N\_Order, Improvements\_Counter$)
    if no improvement is found in any neighborhood then
        Keep the same order
    end if
    if an improvement is found then
        $New\_N\_Order \leftarrow Descending\_Order(N\_Order, Improvements\_Counter)$
        $N\_Order \leftarrow New\_N\_Order$
    end if
    return $N\_Order$
end procedure
```

An example of the adaptive schemes is provided in the following pseudo-code of the Adaptive $GVNS_{pVND}$ algorithm.

Algorithm 5 $AGVNS_{pVND}$

```
procedure $AGVNS_{pVND}(S, k_{\max}, l_{\max}, max\_time, N\_Order, Improvements\_Counter)$
  while $time \leq max\_time$ do
    $S^* = Shake(S, k)$
    $N\_Order \leftarrow Adaptive\_Order(N\_Order, Improvements\_Counter)$
    $S' = pVND(S^*, l_{\max})$
    if $f(S') < f(S)$ then
      $S \leftarrow S'$
    end if
  end while
  return $S$
end procedure
```

4 Computational analysis and results

4.1 Computing environment and parameter settings

The proposed methods have been implemented in Fortran (Intel Fortran compiler 18.0 with optimization option/O3) and ran on a desktop PC running Windows 7 Professional 64-bit with an Intel Core i7-4771 CPU at 3.5 GHz and 16 GB RAM. The execution time limit for the proposed algorithms was set at 60 s. The PLIRP was also modeled in GAMS (GAMS 24.9.1) [5] and its problem instances were solved using CPLEX 12.7.1.0 solver with the time limit of 2 h for the small-sized instances and 5 h for medium and large-sized instances. It should be mentioned that CPLEX ran in the same computing environment with Intel Fortran compiler.

4.2 Computational results on PLIRP instances

In this work 30 new PLIRP instances have been created by following the format of instances proposed in a previous work [39]. They are reported in the form X–Y–Z, where X represents the number of potential depots, Y the number of customers and Z is the number of time periods. These instances are available in: http://pse.cheng.auth.gr/index.php/publications/benchmarks. Table 5 provides the average values of total costs of all the 30 problem instances and for each solution method, using different $k_{\max}$ values. The first sub-column of the main columns 2–4 in Table 5 refers to the average performance of algorithms, while the second one focuses on the best-found solutions.

According to the reported solutions, the parameter value $k_{\max} = 5$ produces in average the best values for the proposed BVNS, ABVNS and $GVNS_{cVND}$ algorithms. The $AGVNS_{cVND}$ algorithm performs better by using a shaking strength of $k_{\max} = 15$, while the $AGVNS_{pVND}$ algorithm produces better solution using the parameter value $k_{\max} = 20$. The results achieved using the $GVNS_{pVND}$ with the parameter value $k_{\max} = 10$ were the best found solutions in average compared with either the same scheme but with different $k_{\max}$ values or the other schemes (18.5% from BVNS, 7.5%

Table 5 Shaking strength analysis on the performance of proposed methods

Method	$k_{\max} = 5$		$k_{\max} = 10$		$k_{\max} = 12$	
	Avg. TC	Best TC	Avg. TC	Best TC	Avg. TC	Best TC
BVNS	41,542.67	40,643.76	41,630.07	40,397.15	41,547.22	40,483.04
ABVNS	37,684.82	36,062.44	37,896.45	35,926.57	38,092.11	36,339.28
$GVNS_{pVND}$	35,304.38	34,455.14	**35,055.97**	**34,295.98**	35,403.77	34,559.92
$AGVNS_{pVND}$	35,735.40	34,426.16	36,031.34	34,625.69	36,012.60	34,674.18
$GVNS_{cVND}$	37,261.36	36,487.80	37,395.34	36,570.37	37,281.85	36,348.84
$AGVNS_{cVND}$	37,687.23	36,361.82	37,715.74	36,416.86	37,570.39	36,375.08

Method	$k_{\max} = 15$		$k_{\max} = 18$		$k_{\max} = 20$	
	Avg. TC	Best TC	Avg. TC	Best TC	Avg. TC	Best TC
BVNS	41,549.48	40,407.81	41,495.84	40,681.73	41,724.02	40,557.38
ABVNS	38,227.76	36,312.05	38,184.58	36,627.57	38,504.86	37,158.58
$GVNS_{pVND}$	35,473.07	34,574.27	35,310.72	34,520.85	35,684.59	34,572.27
$AGVNS_{pVND}$	35,783.10	34,608.32	35,814.60	34,565.73	35,735.4	34,563.19
$GVNS_{cVND}$	37,373.07	36,510.59	37,283.29	36,549.10	37,431.34	36,474.75
$AGVNS_{cVND}$	37,522.14	36,331.10	37,784.85	36,233.78	37,717.24	36,290.09

from ABVNS, and 6.3% from $GVNS_{cVND}$, 7% from $GVNS_{cVND}$ and 1.94% from $AGVNS_{pVND}$).

From a problem size perspective, the $AGVNS_{pVND}$ algorithm produces better solutions for small-sized instances (using $k_{\max} = 20$) and medium-sized instances (using $k_{\max} = 5$) than other approaches, while the $GVNS_{pVND}$ algorithm using the shaking strength $k_{\max} = 10$ is more efficient than other methods for the solution of large problem cases.

From this analysis it is noticed that classic GVNS-based methods provide better solutions than their corresponding adaptive variants. An explanation of this conspicuous observation is that using the adaptive re-ordering mechanism leads to further time consumption. Thus, the number of iterations of the improvement phase is significantly decreased for the case of large-sized instances.

The $GVNS_{pVND}$ uses the Speed Selection Procedure after each local search operator. Table 6 illustrates potential difference between this GVNS scheme and $GVNS_{pVND}$ which applies the SSP once after the completion of a *pVND* iteration. The initial scheme is called $GVNS_{pVND}$_1 and the second one, $GVNS_{pVND}$_2.

Despite the fact that, the $GVNS_{pVND}$_2 scheme produces more best values than the $GVNS_{pVND}$_1, the $GVNS_{pVND}$_1 is slightly better in terms of average solution quality. The previous results demonstrate that a further improved solution method can be proposed by adopting a hybrid scheme. More specifically, for the solution of small- and medium-sized instances the $AGVNS_{pVND}$ method with a shaking strength of $k_{\max} = 20$ will be applied, while the $GVNS_{pVND}$ using $k_{\max} = 10$ will be selected for larger problems. The overall process of the solution method is illustrated in Algorithm 6.

Table 6 Results achieved by the two $GVNS_{pVND}$ schemes on the 30 PLIRP instances

Instance	$GVNS_{pVND}$_1_*Avg*	$GVNS_{pVND}$_1_*Best*	$GVNS_{pVND}$_2_*Avg*	$GVNS_{pVND}$_2_*Best*
4-9-3	25,336.12	25,131.71	25,142.51	**24,932.36**
4-10-3	20,776.21	**20,649.12**	20,780.24	20,709.14
4-10-5	17,503.66	17,481.13	17,491.73	**17,445.33**
4-12-5	27,127.71	27,069.97	27,133.89	**26,958.58**
4-15-3	16,174.14	16,116.6	16,182.92	**16,070.26**
5-12-3	25,571.71	25,514.4	25,619.41	**25,437.45**
5-15-3	16,430.51	**16,389.54**	16,501.02	16,418.06
5-15-5	20,186.32	**20,058.49**	20,348.43	20,164.08
5-18-3	22,202.75	22,109.4	22,216.44	**22,041.49**
5-20-3	20,023.01	**19,845.57**	20,078.84	19,989.58
6-40-5	24,561.19	**24,116.99**	24,575.04	24,341.58
7-52-5	21,218.05	**20,950.23**	21,270.73	21,159
7-55-7	26,922.81	26,574.72	26,677.95	**26,287.57**
8-60-5	31,514.82	31,076.11	31,428.91	**30,732.43**
8-65-7	48,265.4	47,764.15	48,251.36	**47,599.15**
9-70-5	30,936.66	**30,665.22**	31,419.68	30,691.73
9-75-7	29,260.28	**28,588.11**	30,856.81	29,836.59
9-85-5	28,994.05	**26,784.22**	30,735.91	26,815.18
9-88-7	32,165.91	**31,768.12**	32,015.68	31,844.08
10-90-7	27,532.91	**26,879.52**	27,924.18	27,274.21
15-100-7	15,320.21	15,080.38	15,211.79	**14,990.04**
15-100-10	39,134.05	38,666.61	39,425.27	**38,661.29**
15-120-10	38,945.91	**37,473.08**	41,930.15	40,152.04
20-150-10	41,981.88	40,608.95	41,535.43	**40,381.66**
20-180-12	74,109.8	**73069.77**	75,688.98	74,869.72
25-200-12	75,429.81	72,587.55	74,513.52	**68,146.04**
30-250-10	50,017.53	**47,756.68**	51,379.41	48,755.04
30-270-10	58,450.29	56,708.96	59,376.01	**56,637.84**
35-300-10	73,763.83	71,014.24	71,539.39	**69,361.56**
35-310-12	71,821.7	70,379.8	72,096.53	**68,963.52**
Average	*35,055.97*	34,295.98	35,311.61	**34,255.55**

The bold values indicate the best found solutions

Algorithm 6 *Hybrid*_$GVNS_{pVND}$

```
procedure HYBRID ALGORITHM($max\_time$, $l_{\max}$)
    $S \leftarrow$ Construction_Heuristic
    if small-sized instance then
        $k_{\max} \leftarrow 20$
        $S' \leftarrow AGVNS_{pVND}(S, k_{\max}, max\_time, l_{\max})$
    else if medium-sized instance then
        $k_{\max} \leftarrow 5$
        $S' \leftarrow AGVNS_{pVND}(S, k_{\max}, max\_time, l_{\max})$
    else
        $k_{\max} \leftarrow 10$
        $S' \leftarrow GVNS_{pVND}(S, k_{\max}, max\_time, l_{\max})$
    end if
    return $S'$
end procedure
```

The second column of Table 7 reports the results achieved by GAMS/CPLEX. As it can be noticed, the CPLEX solver can solve only eight out of ten small-sized instances within a time limit of 2 h. No solution, using CPLEX, was found for the rest of the problem instances. In the third column, the average objective values achieved by the *Hybrid_GVNS*$_{pVND}$, while the fourth column contains the best results achieved by the *Hybrid_GVNS*$_{pVND}$ algorithm. The last two columns provide the solution gaps between CPLEX and *Hybrid_GVNS*$_{pVND}$. According to the reported results, the *Hybrid_GVNS*$_{pVND}$ provides 2.4% better results than CPLEX (approximately 3% focused on the best found solutions of the *Hybrid_GVNS*$_{pVND}$ algorithm). Based on the fact that, the CPLEX solver cannot provide any feasible solution for medium- and large-sized instances even within a time limit of 5 h and its high computational time for finding the reported solutions in the case of the small-sized instances it can be concluded that, the proposed *Hybrid_GVNS*$_{pVND}$ algorithm is an efficient method for solving large-scale PLIRP instances.

Table 8 reports the best known solutions of the 30 PLIRP instances.

Table 9 summarizes the number of the vehicles used for each problem instance using the *Hybrid_GVNS*$_{pVND}$ algorithm. According to the number of opened depots, all the proposed methods open exactly two depots in each problem instance. Based on the total demand of customers, two depots are the minimum required for fulfilling customers demands per instance. A replenishment policy highly impacts the produced solutions. Furthermore, a flexible replenishment policy enables the building of cost efficient routing patterns [37,39].

Figure 13 illustrates the effect of flexible replenishment policy on the routing and the inventory costs. More specifically, the values of routing and inventory costs reported in the successful iterations of the *GVNS*$_{pVND}$*_1* are depicted. Routing costs can be decreased by allowing flexible reorder points and order quantities for the customers due to the reduction of deliveries or the efficient clustering of customers into routes. For instance, the deliveries to a distant customer can be reduced by replenishing it with more product quantities in less time periods. Also, this shifting of product quantities results on more available space in the vehicles, so more customers can be serviced by the same vehicle. This can lead to cost efficient routing circuits. On the other hand, the deferred deliveries leads to an increase in the inventory cost.
It should be mentioned that, the solutions obtained by the *Hybrid_GVNS*$_{pVND}$ algorithm do not always follow the previous relation. In some cases, the algorithm splits the routes in order to reduce the routing cost without increasing the inventory cost. However, the splitting strategy leads to the usage of more vehicles.

4.3 Sensitivity analysis

To further consider the significance of the flexible replenishment policy, the effect of holding costs on the total cost is studied through a parametric analysis. Holding costs are crucial for the performance of logistics design and operation. In the literature, several works studied the effect of holding cost variations on the overall performance of logistic systems [2,17]. In this work, two testing scenarios are considered. In the first one, a holding cost increase by 10% is examined, while in the second one the

Table 7 Comparative analysis between CPLEX and $Hybrid\_GVNS_{pVND}$ on the small- and medium-sized instances

Instance	CPLEX (a)	$Hybrid\_GVNS_{pVND}\_Avg$ (b)	$Hybrid\_GVNS_{pVND}\_Best$ (c)	$Gap(b-a)$ (%)	$Gap(c-a)$ (%)
4-9-3	25,182.74	25,065.69	24,775.13	0.46	1.62
4-10-3	19,908.16	20,787.31	20,619.48	− 4.42	− 3.52
4-10-5	17,786.65	17,502.38	17,457.5	1.6	1.85
4-12-5	26,741.55	27,053.31	26,958.67	− 1.17	− 0.81
4-15-3	15,370.4	16,125.36	15,968.86	− 4.91	− 3.89
5-12-3	25,353.81	25,517.97	25,451.99	− 0.65	− 0.39
5-15-3	18,670.19	16,422.19	16,406.91	12.04	12.12
5-15-5	N/A	18,224.79	18,084.47	–	–
5-18-3	26,353.47	22,078	21,982.96	16.22	16.58
5-20-3	N/A	20,124.22	19,878.09	–	–
6-40-5	N/A	24,872.74	24,090.17	–	–
7-52-5	N/A	21,093.97	20,810.7	–	–
7-55-7	N/A	26,702.27	26,470.52	–	–
8-60-5	N/A	31,436.84	31,249.98	–	–
8-65-7	N/A	47,497.09	46,207.73	–	–
9-70-5	N/A	30,978.96	30,310.59	–	–
9-75-7	N/A	29,266.82	28,533.75	–	–
9-85-5	N/A	27,883.9	27,014.67	–	–
9-88-7	N/A	31,860.02	31,750.8	–	–
10-90-7	N/A	27,129.59	26,708.53	–	–

Table 8 Best known solutions for the 30 PLIRP instances

Instance	BKS	Instance	BKS
4-9-3	24,775.13	9-70-5	30,310.59
4-10-3	20,619.48	9-75-7	28,533.75
4-10-5	17,457.5	9-85-5	27,014.67
4-12-5	26,958.67	9-88-7	31,750.8
4-15-3	15,968.86	10-90-7	26,708.53
5-12-3	25,451.99	15-100-7	15,080.38
5-15-3	16,406.91	15-100-10	38,666.61
5-15-5	20,012.77	15-120-10	37,473.08
5-18-3	21,982.96	20-150-10	40,608.95
5-20-3	19,878.09	20-180-12	73,069.77
6-40-5	24,090.17	25-200-12	72,587.55
7-52-5	20,810.7	30-250-10	47,756.68
7-55-7	26,470.52	30-270-10	56,708.96
8-60-5	31,249.98	35-300-10	71,014.24
8-65-7	46,207.73	35-310-12	70,379.8

Table 9 Number of used vehicles per instance using $Hybrid\_GVNS_{pVND}$

Instance	#Vehicles	Instance	#Vehicles	Instance	#Vehicles
4-9-3	4	6-40-5	7	15-100-7	4
4-10-3	5	7-52-5	4	15-100-10	4
4-10-5	3	7-55-7	3	15-120-10	4
4-12-5	3	8-60-5	13	20-150-10	5
4-15-3	3	8-65-7	28	20-180-12	9
5-12-3	6	9-70-5	8	25-200-12	8
5-15-3	4	9-75-7	3	30-250-10	4
5-15-5	10	9-85-5	5	30-270-10	5
5-18-3	8	9-88-7	4	35-300-10	11
5-20-3	5	10-90-7	3	35-310-12	4

holding cost is increased by 15%. The $Hybrid\_GVNS_{pVND}$ algorithm, which marked as the most efficient between the proposed solution methods, was used for solving the problem instances in these two scenarios. Table 10 provides both the average and best found results per scenario. To clarify, the terms $OV\_Avg\_X\%$ and $OV\_Best\_X\%$ refer to the Objective Value in average or best case (respectively), under a holding cost variation of $X\%$.

The results indicate that, the second scenario produces 0.45% worse solutions in average comparing to the first one. Moreover, the initial average solutions of the instances are 1.98% and 2.44% better than those achieved in the first and the second scenario respectively, which means that the objective value seems to be sensitive on

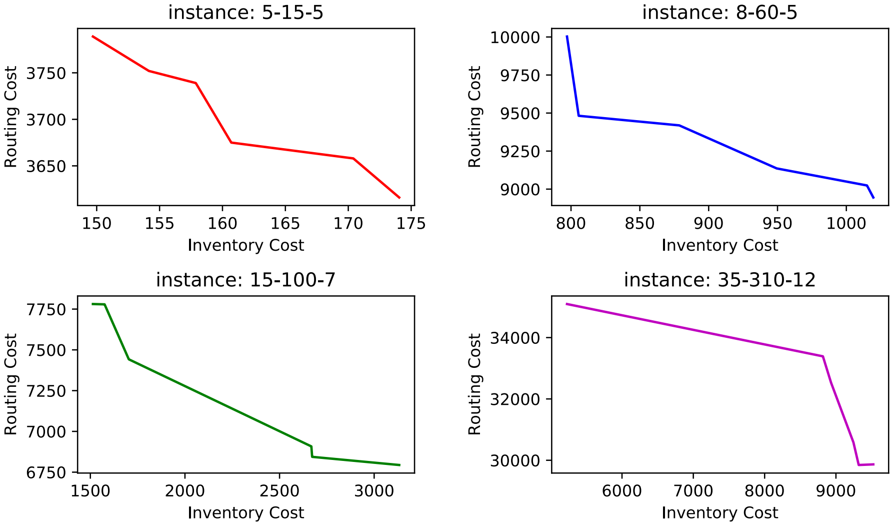

Fig. 13 Effect of flexible replenishment policy on the relationship between routing and inventory costs

the changes of the holding costs. However, the adoption of a flexible replenishment policy keeps the increase of the cost in relatively low levels.

Also, there are some instances in which better solutions were produced by increasing the holding costs. This is justified as, an increase in the holding costs, forces the usage of more vehicles (max two more than the reported vehicles in Table 9) in order to form better routing patterns, thus leading to further routing cost reduction.

5 Conclusions

This work presents a new green supply chain network optimization problem, which considers both economic and environmental concerns. GVNS- and Adaptive GVNS-based solution approaches have been developed for the efficient solution of medium- and large-sized instances. A new set of 30 random generated instances is used in extended numerical analyses. A computational $k_{\max}$ parameter analysis has been made. This analysis indicates that, the GVNS scheme which uses the pipe VND ($GVNS_{pVND}$) method in its improvement process is proved to be the most efficient method. In addition, the effect of executing the Speed Selection Procedure either after each local search operator or in the end of each VND iteration is tested. The GVNS scheme which uses the Speed Selection Procedure after each local search operator proved as the most efficient solution method. However, from a problem size perspective, a hybrid solution approach, which uses the Adaptive GVNS ($AGVNS_{pVND}$) algorithm with different $k_{\max}$ values for the solution of small- and medium-sized instances and the $GVNS_{pVND}$ for solving large problem cases. This hybrid solution scheme is compared with the CPLEX solver in ten small-sized instances. The proposed solution method produces approximately 3% better solutions than CPLEX. Finally, a sensitivity analysis is per-

Table 10 The average and best found results in sensitivity analysis scenarios

Instance	*OV_Avg_*10%	*OV_Best_*10%	*OV_Avg_*15%	*OV_Best_*15%
4-9-3	25,181.77	25,041.14	25,188.46	25,031.23
4-10-3	20,863.64	20,781.04	20,848.53	20,793.92
4-10-5	17,520.59	17,432.05	17,525.54	17,475.08
4-12-5	27,082.38	26,984.32	27,107.96	27,029.25
4-15-3	16,126.86	16,044.12	16,119.94	15,980.26
5-12-3	25,598.02	25,485.53	25,519.32	25,485.46
5-15-3	16,435.71	16,399.19	16,435.47	16,419.09
5-15-5	20,285.74	19,992.44	20,357.95	20,184.11
5-18-3	22,096.39	22,014.08	22,096.62	21,975.62
5-20-3	20,033.03	19,932.21	20,021.84	19,940.18
6-40-5	25,032.92	24,918.66	24,720.14	24,404.99
7-52-5	21,321.34	21,167.21	21,383.7	21,154.03
7-55-7	27,103.04	26,566.88	26,828.2	26,664.36
8-60-5	31,558.35	30,769.91	31,495.14	31,157.83
8-65-7	48,084.16	47,074.4	48,907.25	48,480.28
9-70-5	31,429	30,582.42	31,426.12	30,949.3
9-75-7	29,717.7	29,010.54	30,748.95	29,495.22
9-85-5	29,476.81	27,049.91	29,153.98	27,418.47
9-88-7	32,205.27	31,682.29	32,245.75	32,156.69
10-90-7	28,073.3	27,728.77	28,111.57	27,222.74
15-100-7	15,949.38	15,378.73	15,627.67	15,129.28
15-100-10	40,619.23	39,295.37	39,715.32	38,962.52
15-120-10	40,926.89	39,351.76	42,157.16	40,537.22
20-150-10	42,974.34	41,330.91	45,801.18	41,101.49
20-180-12	75,943.7	73,496.76	75,164.3	73,597.62
25-200-12	77,760.61	73,258.41	78,961.45	73,650.02
30-250-10	53,034.67	50,785.36	52,581.2	50,516.56
30-270-10	60,394.23	56,942.57	59,424.19	57,147.17
35-300-10	77,476.23	73,685.28	77,152.02	71,780.28
35-310-12	72,246.86	69,485.7	74,525.83	69,823.05
Average	35,751.74	34,655.6	35,911.76	34,722.11

formed to study the effect of the variations of holding costs on the objective value. The results illustrate that, any increase on the holding costs affects the objective value. However, the use of the flexible replenishment policy keeps the increase of total cost in relatively low levels. Some exceptions are noticed by using more vehicles.

Current work focuses on the investigation of an adaptive mechanism for re-ordering the shaking operators in alternative shaking schemes, in order to further improve the efficiency of the proposed methods. Also, other local search operators will be investigated both in the improvement and shaking steps of the algorithm. Finally, the consideration of the average time-to-target may be a useful metric for further performance evaluation of the developed methods.

References

1. Adenso-Diaz, B., Lozano, S., Moreno, P.: How the environmental impact affects the design of logistics networks based on cost minimization. Transp. Res. Part D **48**, 214–224 (2016)
2. Alfares, H., Ghaithan, A.: Inventory and pricing model with price-dependent demand, time-varying holding cost, and quantity discounts. Comput. Ind. Eng. **94**, 170–177 (2016)
3. Barth, M., Younglove, T., Scora, G.: Development of a heavy-duty diesel modal emissions and fuel consumption model. In: Technical Report. UC Berkeley: California Partners for Advanced Transit and Highways, San Francisco (2005)
4. Bektas, T., Laporte, G.: The pollution-routing problem. Transp. Res. Part B **45**, 1232–1250 (2011)
5. Brooke, A., Kendrik, D., Meeraus, A., Raman, R., Rosenthal, R.: GAMS-A user's guide (1998)
6. Cheng, C., Qi, M., Wang, X., Zhang, Y.: Multi-period inventory routing problem under carbon emission regulations. Int. J. Hybrid Inf. Technol. **182**, 263–275 (2016)
7. Cheng, C., Yang, P., Qi, M., Rousseau, L.: Modeling a green inventory routing problem with a heterogeneous fleet. Transp. Res. Part E **97**, 97–112 (2017)
8. Demir, E., Bektas, T., Laporte, G.: A review of recent research on green road freight transportation. Eur. J. Oper. Res. **237**, 775–793 (2014)
9. Derbel, H., Jarboui, B., Bhiri, R.: A skewed general variable neighborhood search algorithm with fixed threshold for the heterogeneous fleet vehicle routing problem. Ann. Oper. Res. **272**, 243–272 (2019)
10. Duarte, A., Mladenovic, N., Sanchez-Oro, J., Todosijevic, R.: Variable neighborhood descent. In: Martí, R., Pardalos, P.M., Resende, M.G. (eds.) Handbook of Heuristics. Springer, Cham (2016)
11. Dukkanci, O., Kara, B.Y., Tolga, B.: The green location-routing problem. Comput. Oper. Res. **105**, 187–202 (2019)
12. Frifita, S., Mathlouthi, I., Masmoudi, M., Dammak, A.: Variable neighborhood search based algorithms to solve a rich k-travelling repairmen problem. Optim. Lett. (2020)
13. Glover, F., Gutin, G., Yeo, A., Zverovich, A.: Construction heuristics for the asymmetric TSP. Eur. J. Oper. Res. **129**, 555–568 (2001)
14. Hansen, P., Mladenovic, N., Brimberg, J., Moreno-Perez, J.: Variable neighborhood search. In: Gendreau M., Potvin J.-Y. (eds) Handbook of Metaheuristics. International Series in Operations Research and Management Science (2019)
15. Hansen, P., Mladenović, N., Todosijević, R., Hanafi, S.: Variable neighborhood search: basics and variants. EURO J. Comput. Optim. **5**, 423–454 (2017)
16. Herran, A., Manuel Colmenar, J., Duarte, A.: A variable neighborhood search approach for the Hamiltonian p-median problem. Appl. Soft Comput. **80**, 603–616 (2019)
17. Hu, W., Toriello, A., Dessouky, M.: Integrated inventory routing and freight consolidation for perishable goods. Eur. J. Oper. Res. **271**, 548–560 (2018)
18. Jabir, E., Panicker, V., Sridharan, R.: Design and development of a hybrid ant colony-variable neighbourhood search algorithm for a multi-depot green vehicle routing problem. Transp. Res. Part D **57**, 422–457 (2017)
19. Jarboui, B., Derbel, H., Hanafi, S., N., M., : Variable neighborhood search for location routing. Comput. Oper. Res. **40**, 47–57 (2013)
20. Javid, A., Azad, N.: Incorporating location, routing and inventory decisions in supply chain network design. Transp. Res. Part E **46**, 582–597 (2010)
21. Karakostas, P., Panoskaltsis, N., Mantalaris, A., Georgiadis, M.: Optimization of CAR T-cell therapies Supply Chains. Comput. Chem. Eng. **139** (2020a)
22. Karakostas, P., Sifaleras, A., Georgiadis, C.: Basic VNS algorithms for solving the pollution location inventory routing problem. In: Variable Neighborhood Search (ICVNS 2018), volume 11328 of LNCS, pp. 64–76. Springer, Cham (2019a)
23. Karakostas, P., Sifaleras, A., Georgiadis, C.: A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing. Comput. Chem. Eng. **126**, 263–279 (2019b)

24. Karakostas, P., Sifaleras, A., Georgiadis, M.: Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem. Expert Syst. Appl. **153** (2020b)
25. Koc, C., Bektas, T., Jabali, O., Laporte, G.: The fleet size and mix pollution-routing problem. Transp. Res. Part B **70**, 239–254 (2014)
26. Koc, C., Bektas, T., Jabali, O., Laporte, G.: The impact of depot location, fleet composition and routing on emissions in city logistics. Transp. Res. Part B **84**, 81–102 (2016)
27. Leenders, B., Velazquez-Martinez, J., Fransoo, J.: Emissions allocation in transportation routes. Transp. Res. Part D **57**, 39–51 (2017)
28. Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S., N., M., : A two phase variable neighborhood search for the multi-product inventory routing problem. Comput. Oper. Res. **52**, 291–299 (2014)
29. Poonthalir, G., Nadarajan, R.: A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP). Expert Syst. Appl. **100**, 131–144 (2018)
30. Rayat, F., Musavi, M., Bozorgi-Amiri, A.: Bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks: a modified AMOSA approach. Appl. Soft Comput. **59**, 622–643 (2017)
31. Reichert, A., Holtz-Rau, C., Scheiner, J.: GHG emissions in daily travel and long-distance travel in Germany—social and spatial correlates. Transp. Res. Part D **49**, 25–43 (2016)
32. Skouri, K., Sifaleras, A., Konstantaras, I.: Open problems in green supply chain modeling and optimization with carbon emission targets. In: Pardalos, P.M., Migdalas, A. (eds.) Open Problems in Optimization and Data Analysis, pp. 83–90. Springer, Berlin (2018)
33. Smiti, N., Dhiaf, M., Jarboui, B., Hanafi, S.: Skewed general variable neighborhood search for the cumulative capacitated vehicle routing problem. Int. Trans. Oper. Res. **27**, 651–664 (2020)
34. Sánchez-Oro, J., Sevaux, M., Rossi, A., Marti, R., Duarte, A.: Improving the performance of embedded systems with variable neighborhood search. Appl. Soft Comput. **53**, 217–226 (2017)
35. Tolga, B., J.F., E., Psaraftis, H., and Puchinger, J., : The role of operational research in green freight transportation. Eur. J. Oper. Res. **274**, 807–823 (2019)
36. Toro, M., Franco, F., Echeverri, M., Guimaraes, F.: A multi-objective model for the green capacitated location-routing problem considering environmental impact. Comput. Ind. Eng. **110**, 114–125 (2017)
37. Zachariadis, E., Tarantilis, C., Kiranoudis, C.: An integrated local search method for inventory and routing decisions. Expert Syst. Appl. **36**, 10239–10248 (2009)
38. Zhalechian, M., Tavakkoli-Moghaddam, R., Zahiri, B., Mohammadi, M.: Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. Transp. Res. Part E **89**, 182–214 (2016)
39. Zhang, Y., Qi, M., Miao, L., Liu, E.: Hybrid metaheuristics solutions to inventory location routing problem. Transp. Res. Part E **70**, 305–323 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.