A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing

Panagiotis Karakostas[a], Angelo Sifaleras[b], Michael C. Georgiadis[a,*]

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

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

Abstract

This work presents a Mixed Integer Programing (MIP) formulation for a new complex NP-hard combinatorial optimization problem, the Location Inventory Routing with Distribution Outsourcing (LIRPDO). Due to its computational complexity, only small problem instances can be solved by exact solvers. Therefore, a General Variable Neighborhood Search (GVNS)-based metaheuristic algorithm for solving large LIRPDO instances is presented. The proposed approach has been tested on 20 new randomly generated LIRPDO instances, 20 existing benchmark LIRP instances from the literature and 30 new large-scale random generated LIRP instances. An extended numerical analysis illustrates the efficiency of the underlying method, leading to acceptable solutions requiring limited computational effort.

Keywords: Logistics Optimization, Metaheuristics, Location, Inventory, Routing

1. Introduction

The development of efficient computational techniques is crucial for optimizing the design and operation of supply chains (You et al.,2009). Supply chain decision makers should adopt integrated design approaches for the optimization of complex supply chain networks (Papageorgiou,2009; Shahin Moghadam et al.,2014;

*Corresponding author
Email address: mgeorg@auth.gr (Michael C. Georgiadis)

Please cite this paper as:

Karakostas P., Sifaleras A., and Georgiadis M. C., "A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing", *Computers & Chemical Engineering*, Elsevier Ltd., Vol. 126, pp. 263-279, 2019.

The final publication is available at Elsevier via https://doi.org/10.1016/j.compchemeng.2019.04.015

Munoz et al.,2015; Barbosa-Povoa and Pinto,2018; Fahimnia et al.,2018). A major challenge, for improving the competitiveness in supply chain networks, is through the simultaneous tackling of strategic, tactical and operational level decisions (Garcia and You,2015). The location-allocation decisions are typically characterized as strategic because of their long term effects (Tsiakis et al.,2001; Georgiadis et al.,2011; Copado-Mendez et al.,2013; Kalaitzidou et al.,2014; Mestre et al.,2015; Brunaud et al.,2018). The inventory control and transportation decisions constitute the domain of tactical decisions, and the routing decisions are mainly related to the operational level decisions (Rafie-Majd et al.,2018). These highly interdepended and interelated decisions should be treated simultaneously in order to reduce total costs and improve the responsiveness of the overall supply chain (Vicente et al.,2015; Aguirre et al.,2018; Hiassat et al.,2017). The literature is rich with a large number of contributions addressing simultaneously two of the decision levels, through the following combinatorial optimization problems:

- Location Routing Problem (LRP) (Cuda et al.,2015; , Drexl and Schneider,2015),
- Inventory Routing Problem (IRP) (Singh et al.,2015; Dong et al.,2017; Etebari and Dabiri,2016),
- Location Inventory Problem (LIP) (You and Grossmann,2009)

Other studies proposed the integration of two of the decision levels, while considering selected priorities of the third level. More specifically, a Multi-depot Location Routing Problem (MDLRP) taking into account inventory costs was considered by Liu and Lee (2003). They developed a two-phase heuristic applied on 144 random generated test instances. A hybrid Tabu Search (TS)/Simulated Annealing (SA) approach was then proposed for solving the same MDLRP (Liu and Lin,2005). Furthermore, a linear programming model incorporating location, routing, and inventory decisions was proposed by Ambrosino and Scutella (2005), but feasible solutions were presented only for the case of the LRP on 12 single-period instances. Max Shen and Qi (2007) presented a nonlinear integer programming model for the integrated supply chain design. A Lagrangian-relaxation based algorithm was developed and its performance was evaluated on several randomly generated test instances. A Location Arc Routing Problem (LARP) with inventory constraints was studied by Riquelme-Rodriguez et al. (2016). They proposed two location constructive algorithms for building initial solutions and an Adaptive Large Neighborhood Search (ALNS) algorithm for further improvement of solutions.

The integrated Location Inventory Routing Problem (LIRP) has received rather limited attention in the literature (Hiassat et al.,2017; Zhang et al.,2014). This problem considers simultaneous all three decisions levels and it is classified as an NP-Hard problem (Javid and Azad,2010). Because of its computational complexity, large scale LIRP instances cannot be solved to optimality by exact solution methods (Eskandarpour et al.,2017). In order to overcome such computational limitations, heuristic and meta-heuristic approaches are often applied. The first attempt to tackle simultaneously location, inventory and routing decisions was presented by Javid and Azad (2010). They proposed a Mixed Integer Non Linear Programming (MINLP) model and a hybridization of Tabu Search (TS) and Simulated Annealing (SA) for solving large sized problem instances.

Tavakkoli-Moghaddam et al. (2013) presented an MINLP model for a stochastic distribution network and solved five examples using the Lingo software. A hybrid Variable Neighborhood Descent (VND)-Iterated Local Search (ILS) metaheuristic solution approach was applied by Guerrero et al. (2013) for solving LIRP cases, described by an MIP formulation. Reza Sajjadi et al. (2013) developed an MINLP model for the two-layer multi-product capacitated location routing inventory problem (MPCLRIP) and they solved larger instances using a sequential heuristic. Seyedhosseini et al. (2014) developed an MINLP model for the three-level SCN design and solved random generated instances with the Lingo solver for the small cases and a GA for larger instances. A two-stage hybrid TS heuristic for solving the combined LIRP in B2C e-Commerce distribution system was proposed by Chen et al. (2014). Nekooghadirli et al. (2014) studied a bi-objective LIRP and applied four evolutionary based metaheuristics for solving several test instances. Zhang et al. (2014) presented an MIP model for the multi-period LIRP with flexible replenishment policy and they developed a hybrid SA metaheuristic for solving the proposed problem. Liu et al. (2015) studied a stochastic LIRP for designing a logistic system for e-commerce and implemented a Pseudo Parallel hybridization of GA and SA.

Zhalechian et al. (2016) presented an MINLP model for a sustainable closed-loop LIRP and they applied a hybrid two-phase stochastic-possibilistic programming method within a game theory approach, in order to manage the uncertainty and a Self-Adaptive GA for addressing efficient solutions on large instances. A hybrid SA and Imperialist Competitive Algorithm (ICA) for tackling the LIRP was presented by Ghorbani and Akbari Jokar (2016). Hiassat et al. (2017) proposed evolutionary based optimization metaheuristics and more precisely, different versions of GA based solution approaches. Rafie-Majd et al. (2018) studied the design of a Supply Chain (SC) system of perishable products under uncertainty and they employed a Lagrangian Relaxation heuristic for solving it. Table 1 summarizes the key contributions from the LIRP literature. It should be highlighted that, in the fourth column of Table 1 the term "Single*" refers to a perishable single product.

Table 1: Key literature contributions on LIRP

Reference	**P.T.**[1]	**D.T.**[2]	**C.T.**[3]	**R.P.**[4]	**F.C.**[5]	**Model**	**S.M.**[6]
[25]	Single	Stochastic	Single	(Q,R)	Homogeneous	MINLP	Metaheuristic
[43]	Single	Stochastic	Single	(Q,R)	Heterogeneous	MINLP	Exact
[21]	Multiple	Deterministic/Variable	Single	Order up to level	Homogeneous	MIP	Metaheuristic
[38]	Single	Deterministic	Multiple	Fixed order	Heterogeneous	MINLP	Heuristic
[40]	Single	Stochastic	Single	(Q,R)	Homogeneous	MINLP	Exact/Metaheuristic
[7]	Single	Fuzzy	Single	(T,R) Periodic	Homogeneous	MIP	Metaheuristic
[35]	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	MINLP	Metaheuristic
[50]	Multiple	Deterministic/Variable	Single	Flexible	Homogeneous	MIP	Metaheuristic
[27]	Single	Stochastic	Single	(Q,R)	Homogeneous	MINLP	Metaheuristic
[49]	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	MMINLP	Metaheuristic
[19]	Multiple	Deterministic/Variable	Multiple	(Q,R)	Homogeneous	MIP	Metaheuristic
[23]	Multiple	Deterministic/Variable	Single*	-	Homogeneous	MIP	Metaheuristic
[37]	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	MINLP	Heuristic
This work	Multiple	Deterministic/Variable	Single	Flexible	Homogeneous	MIP	Metaheuristic

[1] *Period Type*, [2] *Demand Type*, [3] *Commodity Type*, [4] *Replenishment Policy*, [5] *Fleet Composition*, [6] *Solution Method(s)*

Several companies understood the importance of the strategic relationships and started adopting logistics outsourcing as a key strategic component, in order to increase their competitiveness (Turkay et al.,2004; Hjaila et al.,2016). Cost reduction, decreased service times and improved customer service are considered as the main advantages of logistics activities outsourcing in the literature (Basligil et al.,2011; Zhu et al.,2017). Because of the crucial effect of the decisions integration and activities outsourcing on the performance of the supply chain, the combined study of these components seems to be highly promising.

Table 2 presents the major differences between the proposed contribution and the works of Guerrero et al. (2013) and Zhang et al. (2014). The first column shows the compared works. The second column indicates if a work considers distribution outsourcing decisions, while the third column summarizes the biggest instances solved in each work. Finally, the last column shows the methods dependency. The solution approaches of Guerrero et al. (2013) and Zhang et al. (2014) require an exact solver and as such they have used the well-known commercial optimization solver, CPLEX.

Table 2: Major differences of the current work with previous relevant contributions

Work	**Distribution outsourcing**	**Largest solved instance**	**Method dependency**
Guerrero et al. (2013)	-	5 depots - 15 customers - 5 periods	co-operates with commercial solver
Zhang et al. (2014)	-	25 depots - 300 customers - 7 periods	co-operates with commercial solver
This work	✓	120 depots - 680 customers - 12 periods	self-contained solver

This work introduces the Location Inventory Routing Problem (LIRP) with Distribution Outsourcing (LIRPDO) decisions. The underlying problem variant represents a more realistic situation, in which a company needs to outsource its distribution operation, as it cannot afford vehicles acquisition or a customer-specific fleet of vehicles is required. Then, more decisions should be made, such as the selection of the proper vehicles providers and the most efficient allocation of the company's opened depots to the selected providers. The proposed problem is NP-hard, which means that realistic large-sized problem instances cannot be solved by exact methods. Therefore, a Sequential General Variable Neighborhood Search (GVNS) combined with an Inventory Rescheduling Procedure (InvRP) for solving large instances of LIRPDO is proposed. The main research contributions of this work are summarized as follows:

- An MIP formulation for the LIRPDO.
- A ratio-based locations' selection strategy introduced in the first phase of a two phase construction heuristic.
- Application of a GVNS-based solution approach on 20 random generated LIRPDO instances and comparison with the CPLEX solver. The proposed method is also applied on 20 LIRP benchmark instances from the literature).
- The proposed approach is a self-contained solver.
- A new benchmark set with the current largest instances of LIRP in the open literature have been generated and made publicly available.

This 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 solution method. Finally, Section 5 draws up concluding remarks and some thoughts on potential future extensions.

2. Problem Description

The LIRPDO is defined as a three echelon supply chain network with multiple potential vehicles' providers, multiple potential depots and a number of geographically dispersed customers. Each customer has a deterministic period-variable demand of one type commodity. It is also assumed that all potential vehicles' providers own the same type capacitated vehicles, but each of them has a different fixed-contract cost. A customer can be allocated to exactly one opened depot, and each opened depot can be served by exactly one vehicles' provider over the planning horizon. A vehicle is be sent from the location of its provider to the selected depot, in order to load the necessary quantity of product and then will travel through the customers allocated to its route. Finally, the vehicle will return to the location of its owner. Therefore, the routes are formed as provider-depot-customer(s)-provider. The objective in this problem is to minimize the total cost including of location, inventory, routing and outsourcing service costs. The proposed MIP extends the mathematical model proposed by Zhang et al. (2014) by considering distribution outsourcing decisions. To address those decisions, a new set of binary decision variables $PDA_{b,j}$ and a cost component for selecting vehicle providers, $\sum_{b\in B}\sum_{j\in J} PDA_{bj} * fp_b$, are considered. Also, new constraints (5), (6), (7), (8) and (9) have been introduced in the original model.

For the sake of the reader clarity all model sets, parameters and variables contained are summarized in Tables 3, 4, and 5, respectively.

Table 3: 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
B	set of vehicles' providers

Table 4: Parameters of the mathematical model

Parameter	Explanation
f_j	fixed opening cost of depot j
fp_b	fixed-contract cost of selecting provider b
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}	period variable demand of customer i
c_{ij}	travelling cost of locations pair (i, j)
VA_k	the ownership of vehicle k

Table 5: Decision variables of the mathematical model

Variable	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
$PDA_{b,j}$	1 if depot j is served by provider b; 0 otherwise

$$\begin{aligned} \min \sum_{j\in J} f_j y_j + \sum_{i\in I} h_i \sum_{t\in H} \left(\frac{1}{2} d_{it} + \sum_{p\in H, p<t} w_{itp}\,(t-p) + \sum_{p\in H, p>t} w_{itp}\,(t-p+|H|) \right) \\ + \sum_{i\in V}\sum_{j\in V}\sum_{t\in H}\sum_{k\in K} c_{ij} x_{ijkt} + \sum_{b\in B}\sum_{j\in J} PDA_{bj} * fp_b \end{aligned} \tag{1}$$

Subject to

$$\sum_{j\in V} x_{ijkt} - \sum_{j\in V} x_{jikt} = 0 \qquad \forall i \in V, \forall k \in K,\ \forall t \in H \tag{2}$$

$$\sum_{j\in V}\sum_{k\in K} x_{ijkt} \leq 1 \quad \forall t \in H,\ \forall i \in I \tag{3}$$

$$\sum_{j\in V}\sum_{k\in K} x_{jikt} \leq 1 \quad \forall t \in H,\ \forall i \in I \tag{4}$$

$$\sum_{i\in I} x_{jikt} \geq x_{bjkt} \quad \forall j \in J, \forall b \in B, \forall k \in K, \forall t \in H \tag{5}$$

$$x_{bjkt} \leq PDA_{bj} * VA_{kb} \quad \forall b \in B, \forall j \in J, \forall k \in K, \forall t \in H \tag{6}$$

$$x_{bb_1kt} = 0 \quad \forall b, b_1 \in B, \forall k \in K, \forall t \in H \tag{7}$$

$$x_{bikt} = 0 \quad \forall b \in B, \forall i \in I, \forall k \in K, \forall t \in H \tag{8}$$

$$x_{jbkt} = 0 \quad \forall j \in J, \forall b \in B, \forall k \in K, \forall t \in H \tag{9}$$

$$x_{ijkt} = 0 \quad \forall i, j \in J,\ \forall k \in K,\ \forall t \in H,\ i \neq j \tag{10}$$

$$\sum_{i\in I}\sum_{b\in B} x_{ibkt} \leq 1 \quad \forall k \in K, \forall t \in H \tag{11}$$

$$\sum_{i\in I} q_{ikt} \leq Q_k \quad \forall k \in K,\ \forall t \in H \tag{12}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijkt} \leq |S| - 1 \quad \forall k \in K,\ \forall t \in H,\ \forall S \subseteq I \tag{13}$$

$$x_{jikt} \leq z_{ij} \quad \forall j \in J, \forall i \in I, \forall k \in K, \forall t \in H \tag{14}$$

$$\sum_{j \in J} z_{ij} = 1 \quad \forall i \in I \tag{15}$$

$$z_{ij} \leq y_j \quad \forall i \in I,\ \forall j \in J \tag{16}$$

$$\sum_{i \in I} \left(z_{ij} \sum_{t \in H} d_{it} \right) \leq C_j \quad \forall j \in J \tag{17}$$

$$\sum_{u \in I} x_{ujkt} + \sum_{u \in V \setminus \{i\}} x_{iukt} \leq 1 + z_{ij} \quad \forall i \in I,\ \forall j \in J,\ \forall \mathrm{k} \in \mathrm{K}, \forall t \in H \tag{18}$$

$$\sum_{i \in I} \sum_{k \in K} \sum_{t \in H} x_{jikt} \geq y_j \quad \forall j \in J \tag{19}$$

$$\sum_{i \in I} x_{jikt} \leq y_j \quad \forall j \in J,\ \forall k \in K,\ \forall t \in H \tag{20}$$

$$\sum_{p \in H} w_{itp} = d_{it} \quad \forall i \in I,\ \forall \mathrm{t} \in \mathrm{H} \tag{21}$$

$$\sum_{t \in H} w_{itp} = \sum_{k \in K} q_{ikp} \quad \forall i \in I,\ \forall p \in H \tag{22}$$

$$q_{ikt} \leq M \sum_{j \in V} x_{ijkt} \quad \forall i \in I,\ \forall t \in H,\ \forall k \in K \tag{23}$$

$$\sum_{j \in V} x_{ijkt} \leq M q_{ikt} \quad \forall i \in I,\ \forall t \in H,\ \forall \mathrm{k} \in \mathrm{K} \tag{24}$$

$$\mathrm{x}_{ijkt} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in H,\ \forall k \in K \tag{25}$$

$$y_j \in \{0,1\} \quad \forall j \in J \tag{26}$$

$$z_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \tag{27}$$

$$q_{ikt} \leq \min\left\{Q_k, \sum_{p\in H} d_{ip}\right\} \quad \forall i \in I,\ \forall j \in J\ ,\forall k \in K \tag{28}$$

$$w_{itp} \leq d_{ip} \quad \forall i \in I,\ \forall t, p \in H \tag{29}$$

The above MIP model is an extension of the work of Zhang et al. (2014) and considers distribution outsourcing additionally. The objective function minimizes the total cost consisting of facilities opening cost, holding costs per unit of product kept at customers, routing costs and outsourcing costs. However, a short description of them is also provided in this section. Constraints (2) guarantee the equilibrium between the interior and exterior vehicles' flow in each node. Constraints (3) and (4) guarantee that each customer is visited by exactly one vehicle per period. Constraints (5) ensure that if a vehicle is sent from a provider to a depot, it should also be sent from that depot to a customer in a selected time period. Constraints (6), ensure that a vehicle will be moved from a provider to, an allocated to him depot with a vehicle owned by him. Constraints (7), (8), (9) and (10) forbid a vehicle to be moved from provider to provider, from provider to customer, from depot to provider and from depot to depot, respectively. Constraints (11) prevent a vehicle from performing more than one route per period. Constraints (12) impose that the capacity of each vehicle will not be exceeded. The subtour elimination requirements are given in constraints (13). Constraints (14) guarantee that a vehicle will be travelled from a depot to a customer only if that customer is allocated to the depot. Constraints (15) and (16) ensure that a customer must be allocated to exact one depot over the time horizon. Constraints (17) respect the capacity of each depot. Constraints (18) prevent the linking of a customer to a depot, if the customer is not allocated to that depot. A vehicle can be moved from a depot to a customer, only if that depot is opened as imposed by constraints (19) and (20). The total amount of deliveries must be equal to the demand of each customer as it is stated in constraints (21). Constraints (22) guarantee that, the total amount of scheduled deliveries for a customer must be equal to the overall actual deliveries to that customer. If a customer receives a replenishment on a specific time period by a specific vehicle, he should be visited by that vehicle as imposed by constraints (23) and (24).

3. Solution Approach

3.1. Initialization Phase

In order to find a feasible initial solution, a two-phase constructive heuristic has been implemented. Location and allocation decisions are made in the first phase while, inventory-routing decisions are determined in the second phase.

3.1.1. Location/Allocation Strategy

To determine the location and allocation decisions, a ratio-based depots' selection procedure combined with a nearest customer allocation strategy have been developed. In the depots' selection method, the ratio $\frac{fixed\_opening\_cost}{Capacity}$ is initially computed for each candidate depot and then, the depot with the minimum ratio is chosen. In the case that two or more depots have the same ratio, one of them is selected arbitrarily (commonly the first found). Then, for each opened depot the nearest customers' allocation strategy is applied. More precisely, the nearest customer to the opened depot is chosen. If the total demand of this customer is less or equal to the remaining capacity of the depot then, the selected customer is allocated to the depot. This first phase of the constructive heuristic is executed until the allocation of all customers. Also, each opened depot is allocated to a vehicles' provider based on a minimum cost criterion (fixed-contract cost plus the routing cost depicted as the distance between the provider and the depot).

3.1.2. Inventory-Routing Construction

For each time period and each depot, a number of vehicles is selected in order to guarantee demand satisfaction of customers allocated to the current depot. For each selected vehicle an assignment of customers is done based on the limited capacity of the vehicle. In order to build the route of each vehicle, the Random Insertion move is applied (Glover et al.,2001). According to the inventory decisions, the quantity scheduled to be sent to each customer in each time period equals to its corresponding demand in that period.

3.2. Improvement Phase

3.2.1. Neighborhood Structures

Six neighborhood structures are considered for guiding the search during the improvement phase as follows:

Inter-route Relocate (N_1): This local search operator removes customer i from his current route R_i and re-inserts him in a new route R_b, after customer b, in each period. A prerequisite for applying this move is, both customers i and b to be visited by vehicles in the same periods. Routes R_i and R_b, could be allocated either to the same depot or to different depots over the time horizon. If the move violates the capacity of the vehicle in route R_b, a replenishment shifting move is applied. Four possible cases are met for this neighborhood:

- Case 1: R_i and R_b are assigned to the same depot and no violations occur on vehicles capacities.
- Case 2: R_i and R_b are assigned to the same depot and vehicles capacity violations occur.
- Case 3: R_i and R_b are assigned to different depots and no violations occur on vehicles capacities.
- Case 4: R_i and R_b are assigned to different depots and vehicles capacity violations occur.

In case 1 only routing decisions are taken. In the second case, both routing and inventory decisions are made, while in the third case routing and allocation decisions are improved. Finally, in case 4 routing, inventory and allocation decisions are simultaneously addressed. Figures 1 and 2 provide an illustrative example of applying the inter-route relocation of customer $C3$ after customer $C1$ (customers are allocated to the same depot) in a three periods instance.

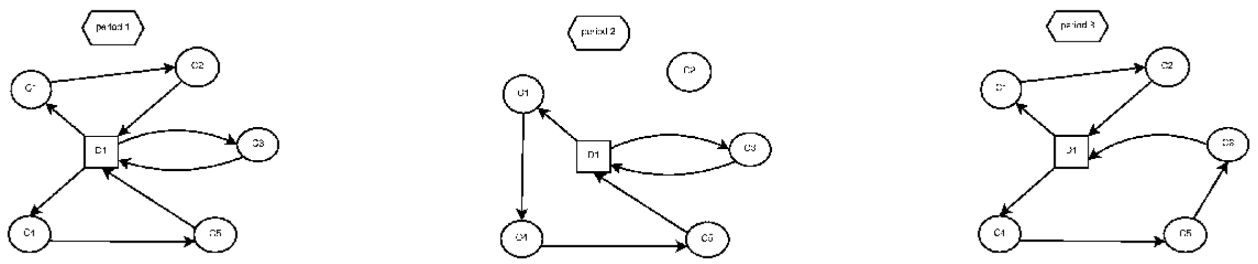

Figure 1: Routes from the same depot in each time period before the application of the inter-route relocate move.

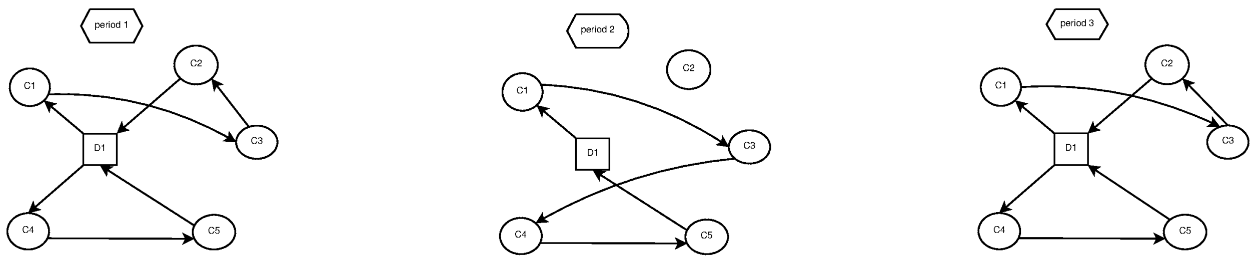

Figure 2: Routes from the same depot in each time period after the application of the inter-route relocate move.

An illustration of the inter-route relocate move applied on customers allocated to different depots, is shown in Figures 3 and 4. More specifically, customer $C2$ is removed from his current position and is inserted in the position after customer $C4$.

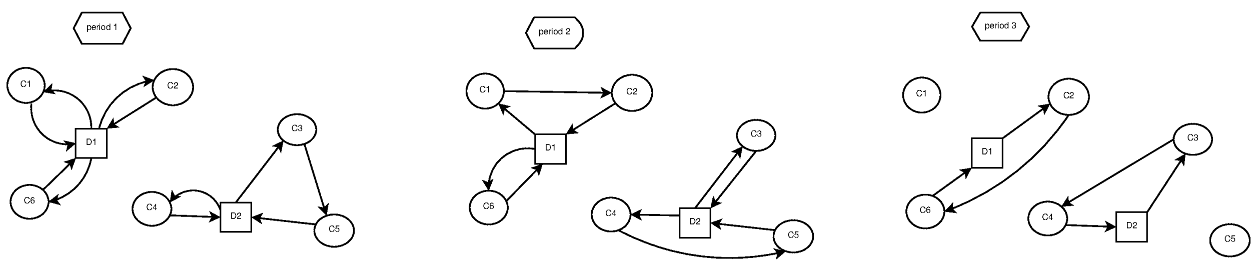

Figure 3: Routes from different depots in each time period before the application of the inter-route relocate move.

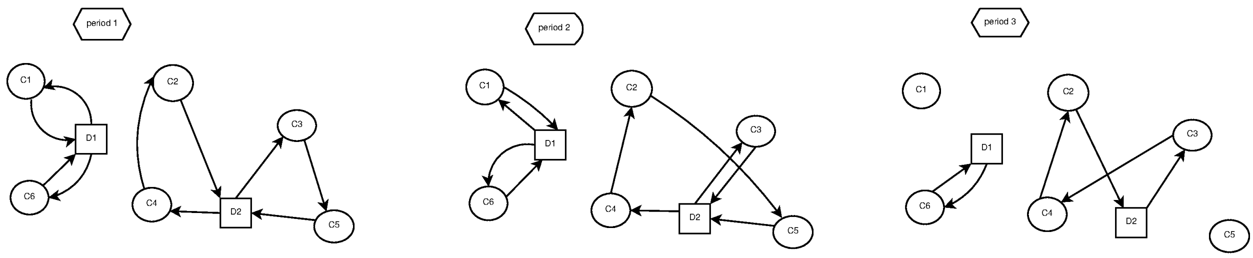

Figure 4: Routes from different depots in each time period after the application of the inter-route relocate move.

Inter-route Exchange (N_2): This neighborhood consists of swapping the positions of two customers (i and b) from different routes (R_i and R_b) over the time horizon. Routes R_i and R_b could be allocated to the same depot or to different depots. In the first case the move may not be applied to all time periods, while in the second case the swapping will be considered as applicable only if it is valid for all time periods. Three special cases could be met by applying this move:

- Case 1: No vehicles' capacity violations occurred.
- Case 2: The demand of customer i violates the capacity of the vehicle servicing customer b in one or more time periods.
- Case 3: The demand of customer b causes violations of the capacity of vehicle servicing customer i in one or more time periods.

In the above case 1 only routing decisions are made, while in cases 2 and 3 both routing and inventory decisions are tackled (inventory: forward/backward shifting to the nearest time periods). If customers are allocated in different depots, changes on allocation decisions are then applied. Figures 5 and 6 illustrate the application of the move in a three period instance. In the first period, customers $C2$ and $C3$ are swapped, while in the periods two and three, the pairs of exchanged customers are ($C2$, $C3$) and ($C3$, $C4$), respectively.

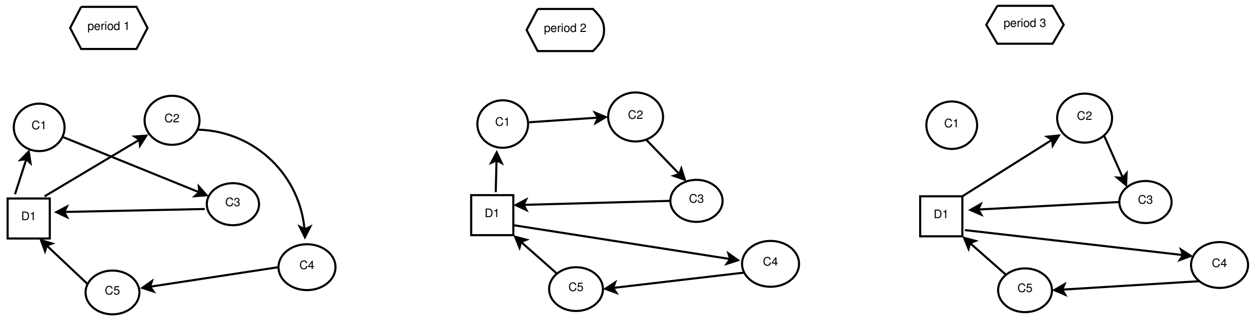

Figure 5: Routes from the same depot in each time period before the application of the inter-route exchange move.

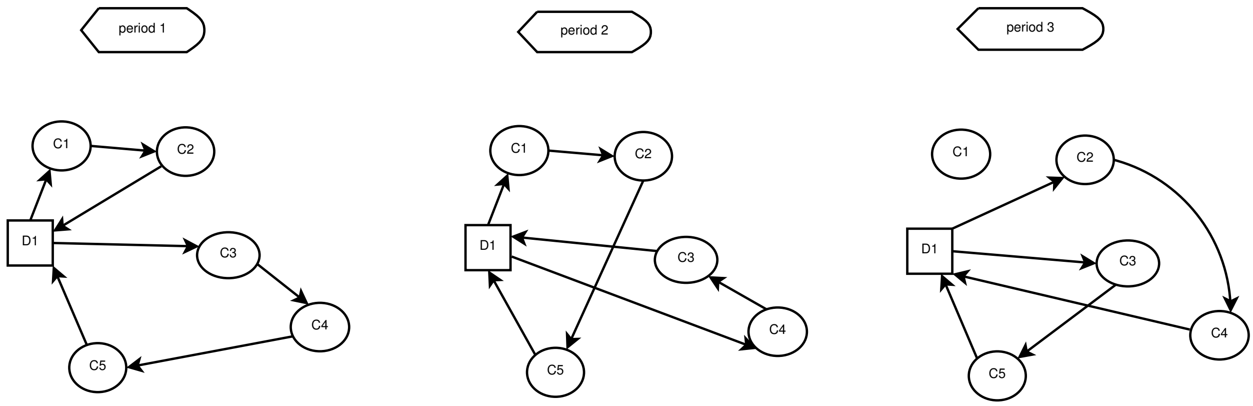

Figure 6: Routes from the same depot in each time period after the application of the inter-route exchange move.

An example of the inter-route exchange move between customers $C3$ and $C4$, allocated to different depots, is illustrated in Figures 7 and 8.

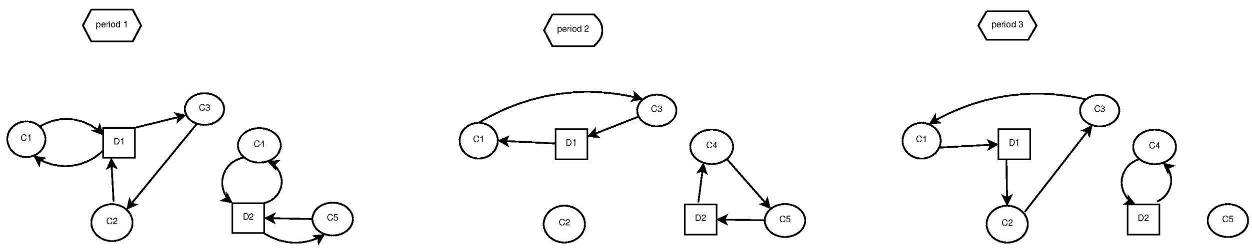

Figure 7: Routes from different depots in each time period before the application of the inter-route exchange move.

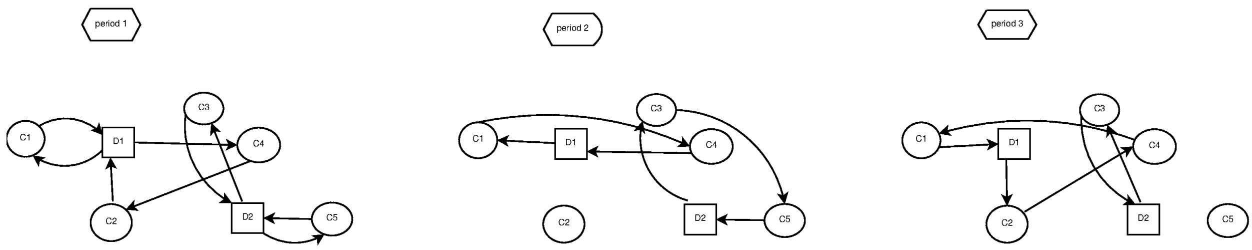

Figure 8: Routes from different depots in each time period after the application of the inter-route exchange move.

Exchange Opened-Closed Depots (N_3): This neighborhood consists of exchanging a closed depot i with a currently opened one j. The exchanging cost is calculated for each closed depot, with all opened depots. Then, the opened depot with the minimum exchanging cost is marked as closed and the validation of the move is examined. In the case of a valid move, a reordering of the routes allocated on depot j is calculated, based on the minimum insertion cost criterion of depot i. If the overall cost (location and routing costs) is decreased then, the move is marked as accepted and it is applied. The move is summarized in the following example in Figures 9 and 10. As it can be seen, in the route of customers $C3, C4$ and $C5$, a routing re-ordering has also been applied.

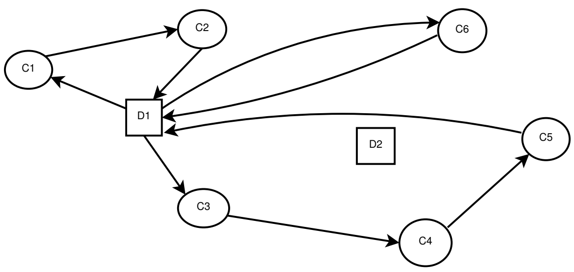

Figure 9: Routes allocation before the opened-closed exchange move.

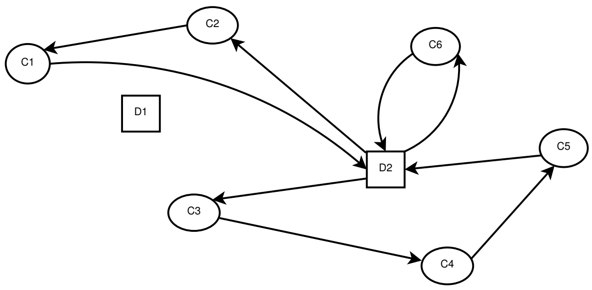

Figure 10: Routes allocation after the opened-closed exchange move.

Intra-route Relocate (N_4): The intra-route relocate operator removes a customer from its current position in its route and re-inserts it in a different position. This move handles only routing decisions. In Figures 11 and 12 the relocation of customer $C2$ after the customer $C4$ is illustrated.

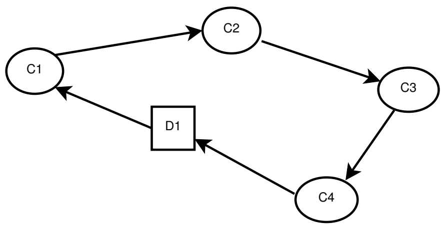

Figure 11: Routes from the same depot in each time period before the application of the intra-route relocate move.

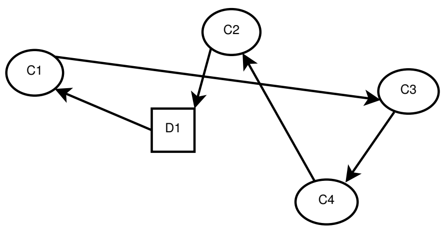

Figure 12: Routes from the same depot in each time period after the application of the intra-route relocate move.

2-2 Replenishment Exchange (N_5): This local search operator randomly selects two time periods t_1 and t_2 and then finds the two most distant customers i and b, both serviced in those two periods. Then, it computes the cost changes of removing i and b from their routes in periods t_1 and t_2 respectively and shifting their receiving deliveries from t_1 to t_2 for customer i and from t_2 to t_1 for b. This move is applied only in case where an improvement is produced and no vehicles' capacities are violated. Figures 13 and 14 provide an illustrative example of this move, applied on customers $C1$ and $C4$, that are allocated in the same route.

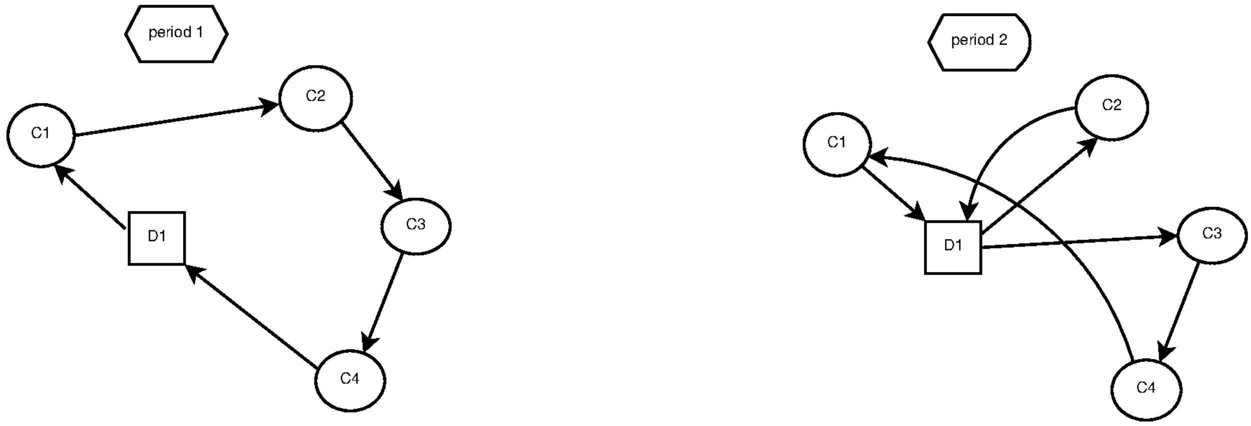

Figure 13: Routes in the two selected time periods before the application of the 2-2 replenishment exchange move.

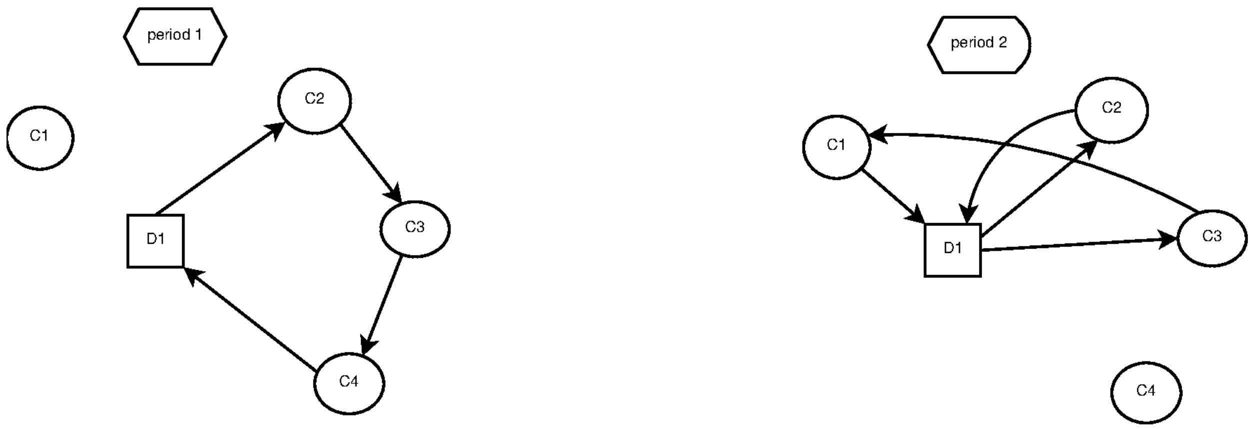

Figure 14: Routes in the two selected time periods after the application of the 2-2 replenishment exchange move.

The 2-2 Replenishment Exchange can also be applied on customers allocated to different routes. An example is presented in Figures 15 and 16, in which the move is applied between customers $C1$ and $C5$.

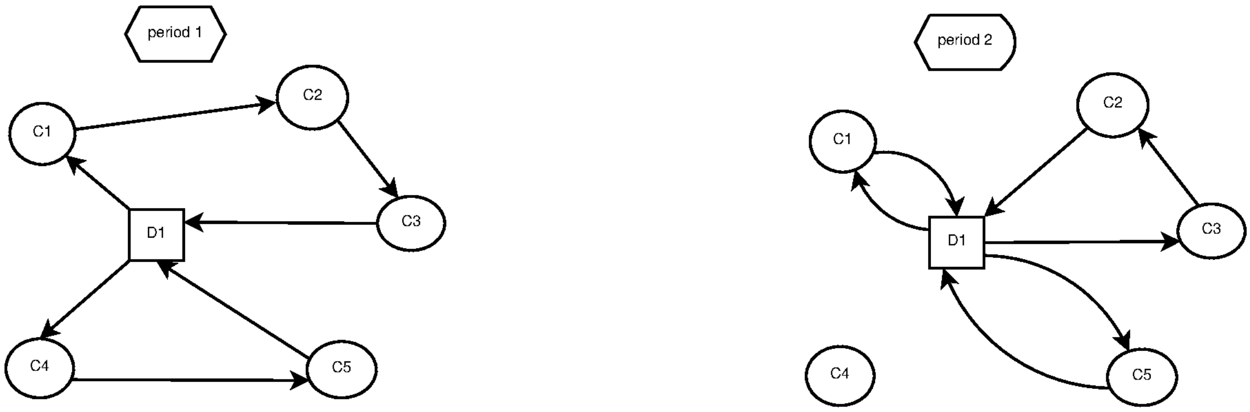

Figure 15: Routes in the two selected time periods before the application of the 2-2 replenishment exchange move.

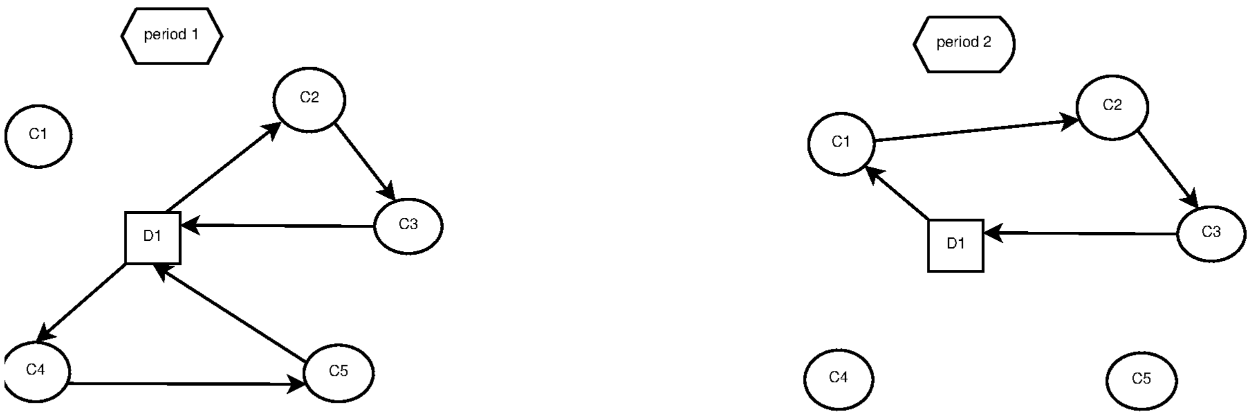

Figure 16: Routes in the two selected time periods after the application of the 2-2 replenishment exchange move.

Change Provider (N_6): This local search operator examines for each opened depot if an improvement may be achieved by allocating it to an other vehicles' provider. In the following illustrated example (Figure 17), the depot $D1$ which is allocated to provider $P1$, it will be allocated to provider $P2$. The order of customers in the routes remains the same.

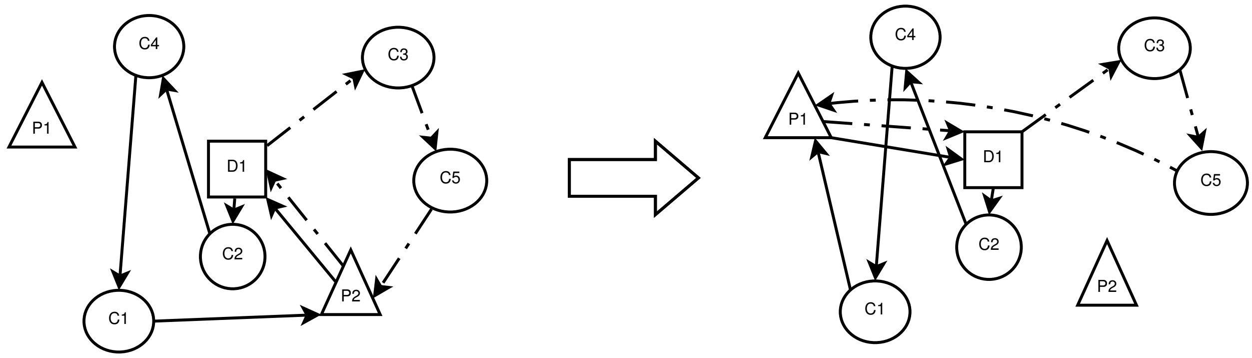

Figure 17: An example of the Change Provider operator

In order to avoid vehicles' capacity violations by applying the Inter-route Relocate and Inter-route Exchange moves, shifting of surplus product quantity may be needed to be also applied. An example of the application of the shifting procedure is given in Figure 18. As it can be seen, the surplus quantity in the second time period is equal to 15 for a selected customer. This customer is also serviced in first and third time periods and the available free space in the corresponding vehicles in these periods are 18 and five respectively. So, five units of product are shifted forward to the third time period and 10 units are shifted backward in the first time period.

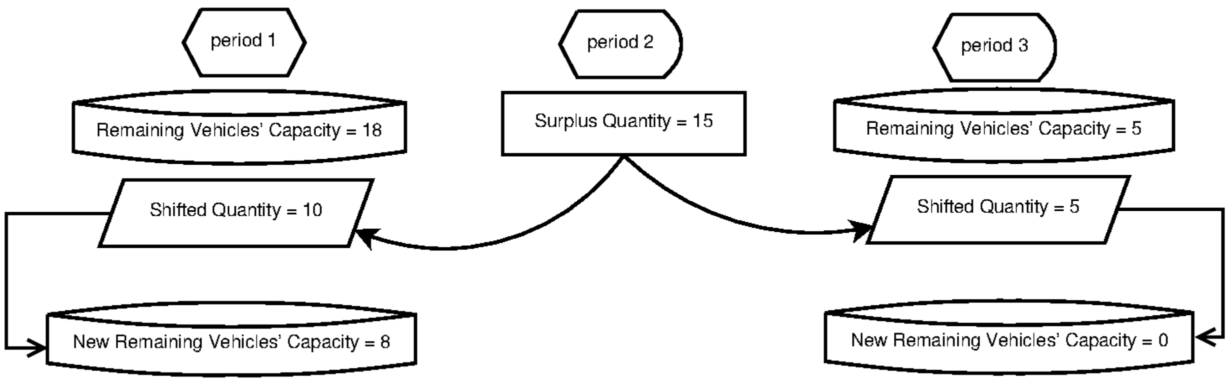

Figure 18: Example of the quantity shifting procedure.

3.2.2. Shaking Procedure

A shaking procedure is developed in order to escape from local optimum solutions (Hansen et al.,2017). Thus, in each shaking phase a number of random jumps are applied in a randomly selected neighborhood from a predefined set of neighborhoods. The pseudo-code of this diversification method is presented in Algorithm 1, with the incumbent solution S and the maximum number of random jumps $k_{max} = 12$ (where k_{max} was experimentally set) as input. The new solution S' is obtained by applying k (where $1 < k < k_{max}$) times one randomly selected neighborhood (from the total $l_{max} = 4$ neighborhoods) and it is then returned as output.

Algorithm 1 Shaking Procedure

```
procedure SHAKE(S, k, l_max)
    l = random_integer(1, l_max)
    for i ← 1, k do
        select case(l)
        case(1)
        S' ← Inter_Relocate(S)
        case(2)
        S' ← Exchange_OpenedClosed_Depots(S)
        case(3)
        S' ← Intra_Relocate(S)
        case(4)
        S' ← Inter_Exchange(S)
        end select
    end for
     Return S'
```

3.2.3. General Variable Neighborhood Search (GVNS)

The Variable Neighborhood Search (VNS) proposed by Mladenovic and Hansen (1997), is a trajectory-based metaheuristic (manage a single solution at each step time) and it is used as a flexible framework for building heuristics (Hansen et al.,2017). The main idea of VNS is the systematic change of predefined neighborhood structures during the search for an optimal or approximately optimal solution. This systematic process is applied as a repeated execution of three basic search ingredients until a stopping criterion is met. These three search steps are (Hansen et al.,2017):

- Shaking Procedure (as a diversification phase for escaping locally optimal solutions).
- Neighborhood Change Step (for guiding purposes while VNS explores the solution space).
- Improvement Procedure (as an intensification phase for improving the incumbent solution).

Several variants of VNS have been already successfully developed for various applications. Four well-known VNS versions are the Basic VNS (BVNS), the Variable Neighborhood Descent (VND), the General VNS (GVNS) and the Reduced VNS (RVNS). BVNS exploits neighborhood structures in both deterministic and stochastic way. The stochastic part of this procedure is utilized through the shaking phase, while the deterministic approach is used in the improvement phase. In VND the exploration of neighborhoods is accomplished in a deterministic way. More specifically, several neighborhood structures can be examined both in a sequential or in a nested way for improving an incumbent solution (Hansen et al.,2017). Actually, the VND approach consists of an improvement procedure and a neighborhood change step. Based on the neighborhood change step rules, four different sequential VND schemes have been proposed in the literature. These schemes are the basic VND (bVND) in which the sequential neighborhood change step is applied, the pipe VND (pVND) which uses the pipe neighborhood change step, the cyclic VND (cVND) that adopts the cyclic exploring method and the union VND (uVND) in which all the predefined neighborhoods are treated as a single one. For further information, the interested reader is directed to a recent survey (Hansen et al.,2017). The GVNS variant is an extension of the BVNS, and its main difference is the usage of a VND scheme as an improvement strategy. In this work, the GVNS-based solution methods use the cyclic and pipe VND as the intensification phase and both the first and best improvement (FI and BI respectively) search strategy are examined. The proposed GVNS algorithm is summarized in the following pseudo-code.

Algorithm 2 General VNS

procedure GVNS($S, k_{max}, max\_time, l_{max}$)
  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

3.2.4. Inventory Rescheduling Procedure

The Inventory Rescheduling Procedure (InvRP) functions as a post optimization part of the proposed solution approach. In each iteration, the most distant customer is selected and the total periods in which this customer is visited by vehicles, in order to satisfy his demand, are kept. Then, an alternative replenishment scheme is examined, trying to reduce the periods needed to visit the selected customer and as a result to reduce the routing costs. The method is terminated either when all customers have been checked or a time stopping criterion is met. An explanation of the variables and the parameters presented in the pseudo-code of Inventory Rescheduling Procedure is firstly given and then, the pseudo-code is provided in Algorithm 3.

- $NPeriods$: Keeps the total number of time periods.
- $NumOfNeededPeriods$: keeps the minimum number of periods needed to satisfy the total demand of a selected customer.
- $DemOfI$: Keeps the total demand of a selected customer over the planning horizon.

- $AvailableVehicles$: a binary 2D array ($N_{Vehicles} * N_{Periods}$) which denotes if a vehicle can visit a selected customer in a period (value equals 1, or not (value equals 0).

- $Veh2ServeI$: Stores the vehicle scheduled to visit a selected customer in a specific period.

- $Period2ServeI$: Logical array which marks the selected periods in the new replenishment plan as "True".

Algorithm 3 Inventory Rescheduling Procedure

procedure INVRP(S)

while a stopping criterion is not met **do**

Find the most distant customer i of all opened depots

Mark customer i as "checked"

for $t \leftarrow 1, NPeriods$ **do**

Find all the available vehicles for visiting i in t

Store those vehicles in $AvailableVehicles$

end for

Compute the total periods in which i is currently serviced, $NPeriodsServedI$

Compute the total demand of customer i, $DemOfI$

$NumOfNeededPeriods = 0$

while $DemOfI > 0$ **do**

Find an unselected vehicle $k \in AvailableVehicles$ with maxCapacity

Keep the vehicle in $Veh2ServeI$ and the period in $Period2ServeI$

$NumOfNeededPeriods = NumOfNeededPeriods + 1$

Recalculate $DemOfI$ based on the partial new replenishment schedule

end while

if $NumOfNeededPeriods < NPeriodsServedI$ **then**

for $t \leftarrow 1, NPeriods$ **do**

if $Period2ServeI(t)$ **then**

Calculate changes on Inventory_Cost and Routing_Cost

Reschedule vehicle routes for i

end if

end for

if $Improvement$ **then**

Renew Inventory and Routing Costs and Inventory_Levels

Apply the routes' rescheduling

end if

end if

end while

return S

3.2.5. GVNS-InvRP

The pseudo-code of the GVNS-InvRP follows in Algorithm 4.

Algorithm 4 GVNS-InvRP

```
procedure GVNS-InvRP
    S ← TwoPhaseConstructionHeuristic
    while time < max_time do
        S' ← GVNS(S)
    end while
    S ← InventoryReschedulingProcedure(S')
    return S
```

Figure 19 illustrates a flowchart of the proposed solution approach.

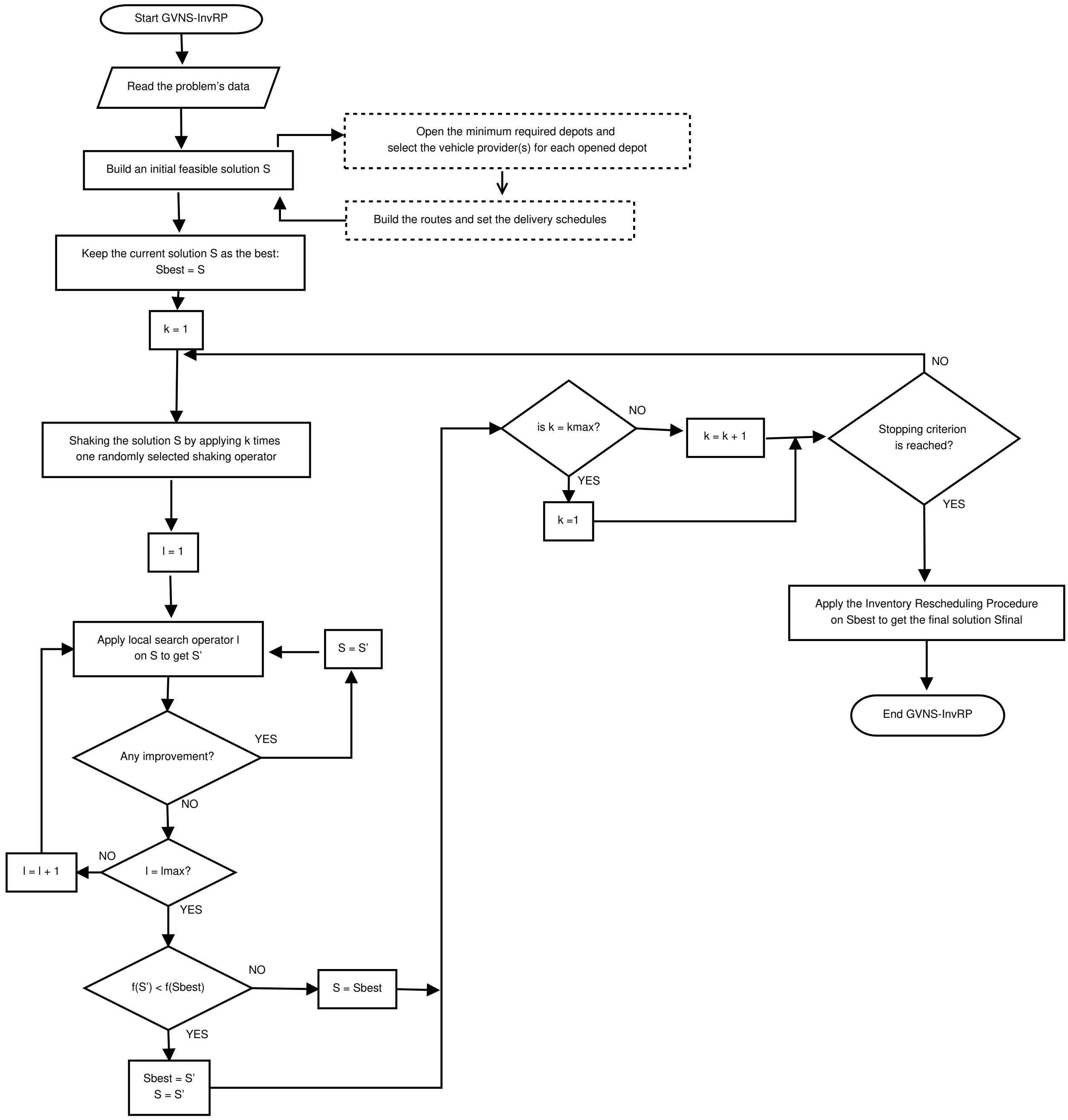

Figure 19: The flowchart of the proposed solution approach.

Initially, a feasible solution is obtained using the two-phase construction heuristic. This is the initial solution of the proposed solution method. GVNS is iteratively executed for 60s and alternates its shaking procedure, the pipe-VND procedure and the solution update step. In each shaking iteration, one of the shaking operators (see subsection 3.2.2) is randomly selected and applied k times (where the parameter k starts with the value one and increased by one in each iteration until $k = kmax$. If k equals k_{max} and the stopping criterion is not met, then the parameter k is set to one). Then, the pVND is applied. In this improvement step, all the local search operators are sequentially applied with the predefined order (see subsection 3.2.1). The search with each local search operator is continued until no more improvements are obtained. After the completion of the pipe-VND, a solution update is performed, by checking if a better solution is available. Subsequently to the GVNS, the Inventory Rescheduling procedure is applied for each customer, starting from the most distant one. The goal of this post-optimization method is to reduce total cost by rescheduling the replenishment plan mainly of the most distant customers, in order to avoid frequent deliveries to them.

Furthermore, an auxiliary subroutine has been developed in order to ensure the feasibility of each solution. This subroutine examines whether the new solution satisfies the constraints of the model and checks the validations of cost renewals.

4. Computational Analysis and Results

In this section, a computational analysis is presented in order to evaluate the performance of the proposed solution method. In subsection 4.1, the necessary technical details (e.g., computing environment) are provided. Subsection 4.2 provides the results achieved by solving the LIRPDO while the subsection 4.3 summarizes the results obtained by the proposed algorithm on 20 LIRP benchmarks from the work of Zhang et al. (2014) and compared with those achieved by the proposed methods presented in the same work. In subsection 4.4 the results achieved by the GVNS-InvRP on 30 randomly generated large-scale instances are presented.

4.1. Computing Environment & Parameter Settings

The methods presented in this work were implemented in Fortran 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 compilation of codes was done using Intel Fortran compiler 18.0 with optimization option /O3. Also, the maximum execution time limit was set (max_time = 60s) for the GVNS approach. The LIRPDO instances were modeled using GAMS (GAMS 24.9.1) (Brooke et al.,1998) and solved using CPLEX 12.7.1.0 solver with specified time limit (2h). CPLEX ran in the same computing environment with Intel Fortran compiler.

4.2. Computational results on LIRPDO instances

This is the first study introducing the LIRPDO, thus there are no previously published test instances in order to compare the efficiency of the proposed solution method. Consequently, 20 new instances were randomly generated following the instructions described in subsection 5.3.1 in the work of Zhang et al. (2014). The instances' names are shaped as X-Y-Z-L, where X is the number of potential depots, Y the number of potential vehicles' providers, Z the number of customers and L the number of time periods. These problem instances are available at: `http://pse.cheng.auth.gr/index.php/publications/benchmarks`.

Figures 20 and 21 illustrates the performance of the proposed solution approaches using either the CVND and the PVND as the main improvement phase and following both the first and best improvement search strategies. It is obvious that the overall performance can be improved by adopting an adaptive mixed search strategy. More specifically, the best improvement search strategy will be applied for instances with up to 90 customers and the first improvement search strategy for the cases with more than 90 customers.

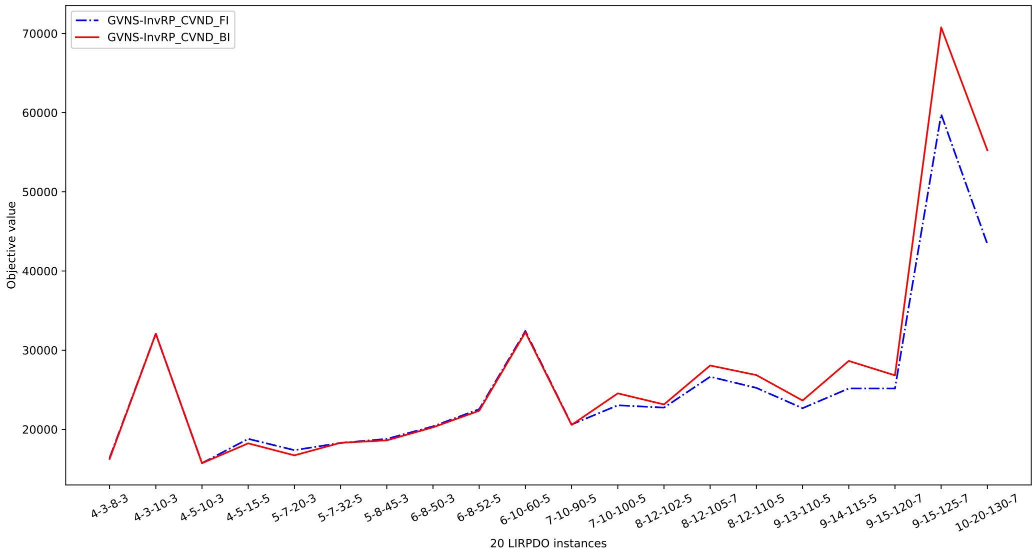

Figure 20: Performance of GVNS-InvRP with CVND using FI and BI.

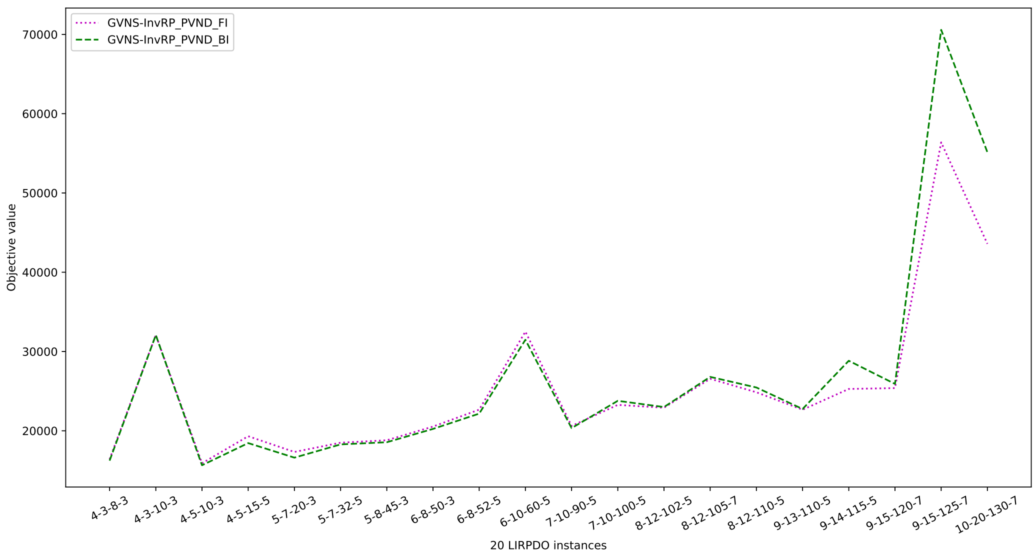

Figure 21: Performance of GVNS-InvRP with PVND using FI and BI.

Table 6 provides the results obtained by CPLEX, GVNS-InvRP with CVND as improvement phase and GVNS-InvRP with PVND. The results of GVNS-InvRP with CVND and GVNS-InvRP with PVND were achieved by the adaptive mixed search strategy. More specifically, in the first column the names of the instances are provided. The second column presents the results achieved by CPLEX, while the third, fourth, and sixth column provide the results achieved by the construction heuristic, the GVNS-InvRP with CVND, and the GVNS-InvRP with PVND, respectively. The fifth and the seventh column show the solution quality gaps of the two proposed methods with the CPLEX. The results of GVNS-based schemes are the average values of five runs per instance.

Table 6: Computational results on 20 LIRPDO

Instance	**CPLEX (a)**	**CH (b)**	**GVNS-InvRP**$_{CVND}(c)$	**gap (a-c) %**	GVNS-InvRP$_{PVND}(d)$	**gap (a-d) %**
4-3-8-3	16,253.89	16,738.29	16,262.33	-0.05	16,235.09	0.12
4-3-10-3	31,509.82	32,125.66	32,073.5	-1.79	32,074.06	-1.79
4-5-10-3	15,727.25	19,409.16	15,724.75	0.02	15,650.88	0.49
4-5-15-5	35,379.9	20,592.56	18,230.54	48.47	18,447.7	47.86
5-7-20-3	-	18,738.69	16,713.07	-	16,611.04	-
5-7-32-5	-	20,475.58	18,299.22	-	18,282.23	-
5-8-45-3	-	21,996.29	18,607.6	-	18,556.2	-
6-8-50-3	-	22,910.67	20,253.66	-	20,235.43	-
6-8-52-5	-	26,446.71	22,348.18	-	22,164.47	-
6-10-60-5	-	42,149.89	32,261.79	-	31,480.11	-
7-10-90-5	-	23,385.28	20,559.74	-	20,336.99	-
7-10-100-5	-	28,507.57	23,036.91	-	23,267.29	-
8-12-102-5	-	25,672.16	22,742.8	-	22,908.09	-
8-12-105-7	-	34,655.32	26,642.79	-	26,560.92	-
8-12-110-7	-	29,469.44	25,236.26	-	24,858.34	-
9-13-110-5	-	24,167.27	22,668.91	-	22,658.02	-
9-14-115-5	-	31,685.59	25,162.47	-	25,271.55	-
9-15-120-7	-	31,560.26	25,156.09	-	25,385.52	-
9-15-125-7	-	71,088.4	59,797.43	-	56,382.8	-
10-20-130-7	-	55,651.04	43,396.27	-	43,584.51	-

Also, in Table 7 the number of the opened depots and selected providers in the final solution of each solution method are provided.

Table 7: The number of opened depots and selected providers on 20 LIRPDO instances.

Instance	**CPLEX**		$GVNS-InvRP_{CVND}$		$GVNS-InvRP_{PVND}$	
	Depots	Providers	Depots	Providers	Depots	Providers
4-3-8-3	2	1	2	1	2	1
4-3-10-3	3	2	3	2	3	2
4-5-10-3	2	1	2	1	2	1
4-5-15-5	3	3	2	1	2	1
5-7-20-3	-	-	2	1	2	1
5-7-32-5	-	-	2	1	2	1
5-8-45-3	-	-	2	1	2	1
6-8-50-3	-	-	2	1	2	1
6-8-52-5	-	-	2	1	2	1
6-10-60-5	-	-	2	1	2	1
7-10-90-5	-	-	2	1	2	1
7-10-100-5	-	-	2	1	2	1
8-12-102-5	-	-	2	1	2	1
8-12-105-7	-	-	2	1	2	1
8-12-110-7	-	-	2	1	2	1
9-13-110-5	-	-	2	1	2	1
9-14-115-5	-	-	2	1	2	1
9-15-120-7	-	-	2	1	2	1
9-15-125-7	-	-	2	2	2	2
10-20-130-7	-	-	2	1	2	1

The CPLEX solver was able to provide an integer solution only for the four small-sized instances (4-3-8-3 to 4-5-15-5). For the next six medium-sized instances CPLEX cannot produce any feasible solution within a 2h time limit, while for the last 10 large-sized problem instances an out of memory error occurred during the execution. Both GVNS-based schemes were able to provide even for the small-sized instances high quality solutions in no more than 60 seconds. More specifically, for the case of the three small-sized instances $4-3-8-3$, $4-3-10-3$ and $4-5-10-3$ the solutions obtained by CPLEX solver in 2h are almost equal to those achieved by the proposed methods in one minute. However, for the case of the instance $4-5-15-5$, both $GVNS-InvRP_{CVND}$ and $GVNS-InvRP_{PVND}$ produce 48.47% and 47.86% better solutions than CPLEX, respectively.

Table 8 reports the best found values of the 20 LIRPDO instances.

Table 8: Best values found on 20 LIRPDO instances

Instance	**Best**	**Instance**	**Best**
4-3-8-3	16,208.14	7-10-90-5	20,310.28
4-3-10-3	32,058.2	7-10-100-5	22,843.81
4-5-10-3	15,501.9	8-12-102-5	22,683
4-5-15-5	18,146.71	8-12-105-7	26,232.41
5-7-20-3	16,604.6	8-12-110-7	24,823.45
5-7-32-5	18,011	9-13-110-5	22,371.2
5-8-45-3	18,537.22	9-14-115-5	24,958.39
6-8-50-3	20,168.32	9-15-120-7	25,005.08
6-8-52-5	22,000.75	9-15-125-7	54,365.64
6-10-60-5	31,410.41	10-20-130-7	42,398.68

4.3. Computational results on LIRP instances (Zhang et al.,2014)

The proposed methods with minor modification (remove the Change Provider local search operator and disable the provider selection in construction heuristic) can also solve LIRP instances, following the approach of Zhang et al. (2014). In this section a comparative analysis between the proposed GVNS-InvRP method, the SA-Hyb-ILRP, and the Sequential heuristic presented by Zhang et al. (2014) is provided. They divided the 20 LIRP instances into small-sized and large-sized instances. The 20 LIRP instances were classified according to their size, as presented in the literature (Mjirda et al.,2014) as follows:

- 10 small sized instances (with less than 20 customers).
- 6 medium sized instances (with customers between 20 and 90).
- 4 large sized instances (with more than 90 customers).

Figure 22 illustrates the performance of the two GVNS-based schemes using the adaptive mixed search strategy in all 20 LIRP instances. The results of the two methods are close enough, except the cases of some instances in which the GVNS-InvRP with the pipe-VND as improvement method generates better solutions. Therefore, the GVNS-InvRP is selected to be compared with the solution approaches presented by Zhang et al. (2014).

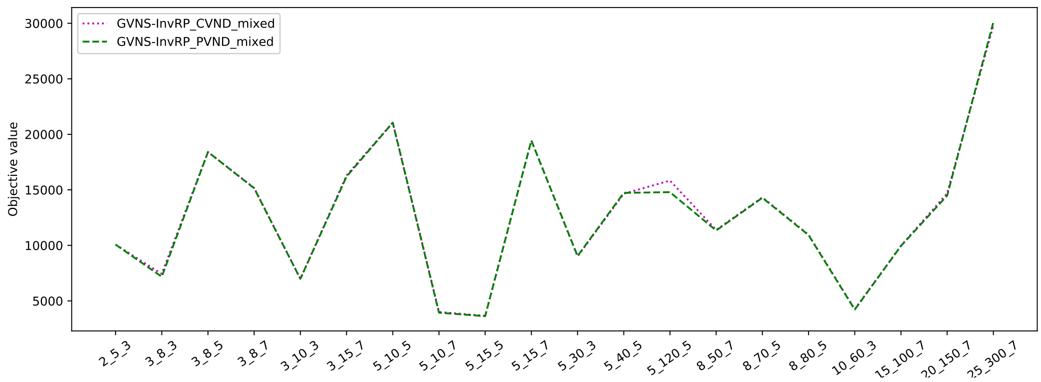

Figure 22: Performance of GVNS-InvRP with CVND and PVND using the adaptive mixed search strategy.

Table 9 shows the results of all three methods. More specifically, in the first column the names of instances are given. The name of each instance formed as $D - C - P$, where D is the number of the potential locations of depots, C is the number of customers and P represents the time periods. In columns 2 and 3 the results of SA-Hyb-ILRP and Sequential heuristic (Zhang et al.,2014) are provided, while the fourth column presents the average values of each instance (average of 5 runs) achieved by GVNS-InvRP method. The last two columns report the solution quality gap between the proposed method and the methods of Zhang et al. (2014).

The gaps are calculated as follows:

$$\text{Gap}_{SA\_Hyb\_ILRP-Seq\_GVNS\_InvRP} = \frac{(S_{SA\_Hyb\_ILRP} - S_{Seq\_GVNS\_InvRP})}{S_{SA\_Hyb\_ILRP}} * 100$$

and

$$\text{Gap}_{Sequentialheuristic-Seq\_GVNS\_InvRP} = \frac{(S_{Sequentialheuristic} - S_{Seq\_GVNS\_-InvRP})}{S_{Sequentialheuristic}} * 100$$

.

Table 9: Computational results on 20 LIRP benchmarks (Zhang et al.,2014)

Instance	**SA-Hyb-ILRP (a)**		**Sequential heuristic (b)**		**GVNS-InvRP (c)**		**gap (a-c) %**	**gap (b-c) %**
	Objective	Time	Objective	Time	Objective	Time		
2-5-3	9,958.98	15.96	10,363.16	1.14	10,072.38	60	-1.14	2.81
3-8-3	6,774.87	46.8	6,799.58	5.1	7,176.86	60	-5.93	-5.55
3-8-5	17,654.66	74.21	19,458.83	6.52	18,407.34	60	-4.26	5.4
3-8-7	14,252.47	125.14	14,372.96	5.25	15,144.31	60	-6.26	-5.37
3-10-3	6,530.9	260.32	7,101.85	12.32	6,986.72	60	-6.98	1.62
3-15-7	15,220.19	485.68	17,980.15	18.65	16,199.62	60	-6.44	9.90
5-10-5	19,936.59	587.63	20,070.7	20.12	21,055.71	60	-5.61	-4.91
5-10-7	3,296.23	495.6	3,709.88	14.23	3,941.18	60	-19.57	-6.23
5-15-5	3,143.41	523.65	4,157.6	17.86	3,622.48	60	-15.24	12.87
5-15-7	18,531.83	547.21	18,820.99	19.85	19,444.16	60	-4.92	-3.31
5-30-3	8,343.27	587.6	8,402.36	18.52	9,043.30	90	-8.39	-7.63
5-40-5	13,507.89	698.52	13,919.62	26.5	14,731.14	90	-9.06	-5.83
5-120-5	28,938.4	1042.68	37,906.5	80.35	14,793.61	90	48.88	60.97
8-50-7	10,127.58	714.3	19,341.65	28.65	11,340.69	90	-11.98	41.37
8-70-5	12,391.97	498.63	12,794.09	16.5	14,284.63	90	-15.27	-11.65
8-80-5	9,520.99	785.6	11,030.4	29.85	10,928.24	90	-14.78	0.93
10-60-3	3,837.94	695.25	4,148.37	20.2	4,216.21	90	-9.86	-1.64
15-100-7	27,761.56	1236.21	37,728.64	30.98	9,947.47	90	64.17	73.63
20-150-7	46,148.96	1562.3	55,912.54	58.47	14,485.69	90	68.61	74.09
25-300-7	87,186.54	2365.87	88,003.22	205.85	30,094.08	90	65.48	65.8
Average	18,153.26	667.46	20,601.15	31.84	12,795.79	75	5.07	14.86

As it is shown in Table 9 the solutions obtained by the GVNS-InvRP on the 20 large-sized instances (Zhang et al.,2014) are 5.07% better than the SA-Hyb-ILRP and 14.86% better than the Sequential heuristic, while the corresponding improvements for the four large-scale instances are 61.9% and 68.63%, respectively. The proposed method is much faster than the SA-Hyb-ILRP as it produces the solutions of all 20 LIRP instances in an average time of 75 s, while the SA-Hyb-ILRP needs 667.46 s. Focusing on the four large-scale instances, the GVNS-InvRP solves them with an average time of 90 s, while the SA-Hyb-ILRP needs an average of 1551.77 s. The Sequential Heuristic functions faster than the GVNS-InvRP for all the 20 LIRP instances, but for the four large-scale instances its average computational time is increased to 93.91 s.

It can be noticed that, the SA-Hyb-ILRP performs 7.64% better than the GVNS-InvRP on small-sized instances and 11.56% on six medium-sized instances. This preeminence of SA-Hyb-ILRP may be attributed to its hybridization with exact methods. Sequential heuristic performs almost equivalently to the GVNS-InvRP for the case of small-scaled instances, with the proposed method to produce 0.72% better solutions and 2.59% better solutions for medium-sized problem instances. This inefficiency of the proposed solution method may be attributed to the use of a predefined order of the local search operators in the pVND, which can potentially limit the exploration performance of the proposed solution method for small- and medium-sized problem instances. However, the main strength of the GVNS-InvRP is its ability of opening the minimum required number of depots for satisfying the total demand of all customers.

Table 10: Number of opened depots in four large-sized LIRP instances (Zhang et al.,2014)

Instance	**CFLP**	**GVNS-InvRP**	**SA-Hyb-ILRP**
5-120-5	2	2	4
15-100-7	2	2	6
20-150-7	2	2	7
25-300-7	2	2	14

Table 10 presents the number of the opened depots by each solution method for the four large-sized instances. The second column provides the number of opened depots in the optimal solution of the Capacitated Fixed-charge Location problem presented by Zhang et al. (2014), and columns 3 and 4 provide the number of opened depots in the final solution of GVNS-InvRP and SA-Hyb-ILRP, respectively. The first column contains the instances names. As it can be observed, GVNS-InvRP focus on opening the minimum number of the needed depots. The randomly opening of depots in Depot-Exchange post optimization operator in SA-Hyb-ILRP might be attributed for opening more depots and as a consequence for increasing the overall cost.

4.4. Computational results on randomly generated large-scale instances

30 new large-scale instances (currently the largest available in the literature) were generated following the instructions described by Zhang et al. (2014) in subsection 5.3.1. The smallest instance consists of 28 depot potential locations, 320 customers, and 7 time periods while the biggest one consists of 120 depot potential locations, 680 customers, and 12 time periods. These problem instances are available `http://pse.cheng.auth.gr/index.php/publications/be`

In order to evaluate the performance of GVNS-InvRP on these large-scale instances, a comparison between GVNS-InvRP and CPLEX is attempted, but an out-of-memory error is occurred. Table 11 reports the average and the best solutions achieved by GVNS-InvRP with either cyclic and pipe VND as improvement method. Each solution reported in Table 11 is the average value of five runs.

Table 11: Computational results on 30 large scale LIRP instances (average & best solutions)

Instance	$GVNS - InvRP_{CVND}$	$GVNS - InvRP_{PVND}$	**Gap %**	**BestKnownValue**
28-320-7	19,335.81	19,431.41	-0.49	19,011.52
30-350-7	18,499.62	17,901.95	3.23	17,882.12
30-375-7	32,852.77	31,379.75	4.48	30,015.27
32-380-7	26,085.52	24,134.91	7.48	23,456.88
35-400-7	20,804.13	21,079.49	-1.32	20,658.84
37-415-7	26,222.85	26,210.01	0.05	25,746.4
40-420-7	21,438.87	21,472.92	-0.16	21,301
42-450-7	29,078.81	29,021.97	0.2	28,740.94
45-480-7	26,407.55	26,398.17	0.04	26,189.98
47-490-7	23,237.25	23,264.26	-0.12	23,015.13
50-490-9	41,778.62	41,851.58	-0.17	41,229
52-495-9	65,948.28	66,398.66	-0.68	65,335.34
55-500-9	31,431.10	31,817.83	-1.23	31,379.49
62-510-9	79,544.16	79,263.62	0.35	78,725.77
65-520-9	32,688.15	32,593.36	0.29	32,072.06
67-540-9	42,637.17	43,064.84	-1	41,954.39
70-550-9	48,195.86	48,273.41	-0.16	48,024
74-560-9	33,359.90	33,081.88	0.83	32,907.79
78-570-9	35,041.01	35,799.88	-2.17	34,862.95
80-580-9	61,245.09	62,322.33	-1.76	60,756.2
85-590-12	68,559.28	69,257.85	-1.02	68,367.91
90-595-12	104,381.54	104,045.38	0.32	103,671.9
92-600-12	78,181.17	78,074.18	0.14	77,596.09
95-610-12	78,976.95	80,957.60	-2.51	76,871.55
98-620-12	65,268.30	64,902.25	0.56	64,808.86
100-650-12	51,805.04	51,429.94	0.72	51,337.58
105-655-12	132,348.48	132,425.78	-0.06	131,798.1
110-660-12	54,551.25	54,447.37	0.19	54,005.45
115-670-12	53,860.97	53,653.04	0.39	52,836.14
120-680-12	61,965.16	61,839.74	0.2	60,905.86
Average	48,857.69	48,859.84	0.22	48,182.15

Table 12: Heuristic vs metaheuristic performance on 30 large scale LIRP instances

Instance	**Two-phase heuristic**	$GVNS - InvRP_{PVND}$	**Gap %**
28-320-7	23,688.94	19,431.41	17.97
30-350-7	20,583.44	17,901.95	13.03
30-375-7	45,948.68	31,379.75	31.71
32-380-7	36,242.85	24,134.91	33.41
35-400-7	27,343.06	21,079.49	22.91
37-415-7	38,818.12	26,210.01	32.48
40-420-7	27,067.15	21,472.92	20.67
42-450-7	38,391.9	29,021.97	24.41
45-480-7	34,203.68	26,398.17	22.82
47-490-7	27,656.52	23,264.26	15.88
50-490-9	58,521.38	41,851.58	28.48
52-495-9	103,278.6	66,398.66	35.71
55-500-9	37,453.72	31,817.83	15.05
62-510-9	138,957.9	79,263.62	42.96
65-520-9	38,588.45	32,593.36	15.54
67-540-9	52,051.25	43,064.84	17.26
70-550-9	61,861.2	48,273.41	21.96
74-560-9	37,137.18	33,081.88	10.92
78-570-9	41,172.02	35,799.88	13.05
80-580-9	88,570.99	62,322.33	29.64
85-590-12	81,680.78	69,257.85	15.21
90-595-12	136,311	104,045.38	23.67
92-600-12	94,972.56	78,074.18	17.79
95-610-12	90,975.61	80,957.6	11.01
98-620-12	75,700.98	64,902.25	14.26
100-650-12	55,773.44	51,429.94	7.79
105-655-12	162,519	132,425.78	18.52
110-660-12	58,072.93	54,447.37	6.24
115-670-12	57,275.8	53,653.04	6.33
120-680-12	67,627.68	61,839.74	8.56
Average	61,948.23	48,859.84	19.84

Many companies face complex supply chain optimization problems, such as the LIRP and the LIRPDO and they try to deal with them using simple heuristics, as the two-phase construction heuristic described in subsection 3.1. However, results reported in Table 12 illustrate clearly that, the use of pure metaheuristic approaches can help companies to improve their cost efficiency. For example, in the case of the large-sized LIRP instances, the $GVNS-InvRPpVND$ has resulted in approximately 20% better solutions (in average) than a simple heuristic.

5. Conclusions

This work considers the optimization of a new complex supply chain problem, the Location Inventory Routing Problem with Distribution Outsourcing (LIRPDO). This problem integrates strategic, tactical, and operational level decisions in order to explore simultaneously their synergistic benefits. Due to its computational complexity, a new metaheuristic solution approach was developed, based on the framework of General Variable Neighborhood Search. An extensive computational analysis on several large-scale problem instances illustrates the efficiency of the proposed approach in terms of solution quality, especially in large-scale problem instances with potential industrial relevance. Two are the main strengths of the proposed solution approach. The first is its ability to open the minimum required number of depots for satisfying the customers demand (see Table 10) . The second one, is the adaptive search strategy, which significantly enhances the performance of the improvement phase in the GVNS component of the proposed solution method. However, the proposed approach does not perform efficiently in small- and medium-sized problem instances.

Future work can focus on investigation of a systematic method to obtain lower bounds for the large-scaled problem instances. The use of relaxation technique such as the Lagrangian Relaxation is envisaged. Moreover, self-adaptive mechanisms can be used within pVND, as an effort to improve its performance. The proposed method can be extended for the solution of more general variants of the problem, as the LIRP with heterogeneous fleet of vehicles, the LIRP with environmental impact considerations, or the consideration of remanufacturing options (Elzakker et al.,2017; Cunha et al.,2017).

References

[1] Aguirre, A., Liu, S., and Papageorgiou, L. (2018). Optimisation approaches for supply chain planning and scheduling under demand uncertainty. Chemical Engineering Research and Design, 138:341–357.

[2] Ambrosino, D. and Scutella, M. (2005). Distribution network design: New problems and related models. European Journal of Operational Research, 105:815–834.

[3] Barbosa-Povoa, A. and Pinto, J. (2018). Opportunities and challenges in sustainable supply chain An operations research perspective. European Journal of Operational Research, 268:399–431.

[4] Basligil, H., Kara, S., Alcan, P., Ozkan, B., and Caglar, E. (2011). A distribution network optimization problem for third party logistics service providers. Expert Systems with Applications, 38:12730–12738.

[5] Brooke, A., Kendrik, D., Meeraus, A., Raman, R., and Rosenthal, R. (1998). GAMS-A Users Guide.

[6] Brunaud, B., Bassett, M., Agarwal, A., Wassick, J., and Grossmann, I. (2018). Efficient formulations for dynamic warehouse location under discrete transportation costs. Computers & Chemical Engineering, 111:311–323.

[7] Chen, D., Sun, G., and Liu, G. (2014). Combined location routing and inventory problem of e-commerce distribution system with fuzzy random demand. International Journal of Hybrid Information Technology, 7:429–442.

[8] Copado-Mendez, P., Blum, C., Guillen-Gosalbez, G., and Jimenez, L. (2013). Large neighbourhood search applied to the efficient solution of spatially explicit strategic supply chain management problems. Computers & Chemical Engineering, 49:114–126.

[9] Cuda, R., Guastaroba, G., and Speranza, M. (2015). A survey on two-echelon routing problems. Computers & Operations Research, 55:185–199.

[10] Cunha, J. O., Konstantaras, I., Melo, R. A., and Sifaleras, A. (2017). On multi-item economic lot-sizing with remanufacturing and uncapacitated production. Applied Mathematical Modelling, 50:772–780.

[11] Dong, Y., Maravelias, C., Pinto, J., and Sundaramoorthy, A. (2017). Solution methods for vehicle-based inventory routing problems. Computers & Chemical Engineering, 101:259–278.

[12] Drexl, M. and Schneider, M. (2015). A survey of variants and extensions of the location-routing problems. European Journal of Operational Research, 241:283–308.

[13] Elzakker, M., Maia, L., Grossmann, I., and Zondervan, E. (2017). Optimizing environmental and economic impacts in supply chains in the FMCG industry. Sustainable Production and Consumption, 11:68–79.

[14] Eskandarpour, M., Dejax, P., and Peton, O. (2017). A large neighborhood search heuristic for supply chain network design. Computers & Operations Research, 80:23–37.

[15] Etebari, F. and Dabiri, N. (2016). A hybrid heuristic for the inventory routing problem under dynamic regional pricing. Computers & Chemical Engineering, 95:231–239.

[16] Fahimnia, B., Davarzani, H., and Eshragh, A. (2018). Planning of complex supply chains: A performance comparison of three meta-heuristic algorithms. Computers & Operations Research, 89:241–252.

[17] Garcia, D. and You, F. (2015). Supply chain design and optimization Challenges and opportunities. Computers & Chemical Engineering, 81:153–170.

[18] Georgiadis, M., Tsiakis, P., Longinidis, P., and Sofioglou, M. (2011). Optimal design of supply chain networks under uncertain transient demand variations. Omega, 39:254–272.

[19] Ghorbani, A. and Akbari Jokar, M. (2016). A hybrid imperialist competitive-simulated annealing algorithm for a multisource multi-product location-routing-inventory problem. Computers Industrial Engineering, 101:116–127.

[20] Glover, F., Gutin, G., Yeo, A., and Zverovich, A. (2001). Construction heuristics for the asymmetric TSP. European Journal of Operational Research, 129:555–568.

[21] Guerrero, W., C., P., Velasco, N., and Amaya, C. (2013). Hybrid heuristic for the inventory location-routing problem with deterministic demand. International Journal of Production Economics, 146:359–370.

[22] Hansen, P., Mladenovic, N., Todosijevic, R., and Hanafi, S. (2017). Variable neighborhood search: Basics and variants. EURO Journal on Computational Optimization, 5:423–454.

[23] Hiassat, A., Diabat, A., and Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. Journal of Manufacturing Systems, 42:93–103.

[24] Hjaila, K., Lainez-Aguirre, J., Zamarripa, M., Puigjaner, L., and Espuna, A. (2016). Optimal integration of thirdparties in a coordinated supply chain management environment. Computers & Chemical Engineering, 86:48–61.

[25] Javid, A. and Azad, N. (2010). Incorporating location, routing and inventory decisions in supply chain network design. Transportation Research Part E, 46:582–597.

[26] Kalaitzidou, M., Longinidis, P., Tsiakis, P., and Georgiadis, M. (2014). Optimal Design of Multiechelon Supply Chain Networks with Generalized Production and Warehousing Nodes. Industrial Engineering Chemistry Research, 53:13125–13138.

[27] Liu, B., Chen, H., Li, Y., and Liu, X. (2015). A pseudo-parallel genetic algorithm integrating simulated annealing for stochastic location-inventory-routing problem with consideration for returns in e-commerce. Discrete Dynamics in Nature and Society.

[28] Liu, S. and Lee, S. (2003). A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration. International Journal of Advanced Manufacturing

Technology, 22:941–950.

[29] Liu, S. and Lin, C. (2005). A heuristic method for the combined location routing and inventory problem. International Journal of Advanced Manufacturing Technology, 26:372–381.

[30] Max Shen, Z. and Qi, L. (2007). Incorporating inventory and routing costs in strategic location models. European Journal of Operational Research, 179:372–389.

[31] Mestre, A., Oliveira, M., and Barbosa-Povoa, A. (2015). Locationallocation approaches for hospital network planning under uncertainty. European Journal of Operational Research, 240:791–806.

[32] Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S., and Mladenovic, N. (2014). A two phase variable neighborhood search for the multi-product inventory routing problem. Computers & Operations Research, 52:291–299.

[33] Mladenovic, N. and Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research, 24:1097–1100.

[34] Munoz, E., Capon-Garcia, E., Lainez-Aguire, J., Espuna, A., and Puigjaner, L. (2015). Supply chain planning and scheduling integration using Lagrangian decomposition in a knowledge management environment. Computers & Chemical Engineering, 72:52–67.

[35] Nekooghadirli, N., Tavakkoli-Moghaddam, R., Ghezavati, V., and Javanmard, S. (2014). Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. Computers & Industrial Engineering, 76:204–221.

[36] Papageorgiou, L. (2009). Supply chain optimisation for the process industries: Advances and opportunities. Computers & Chemical Engineering, 33:1931–1938.

[37] Rafie-Majd, Z., Pasandideh, S., and Naderi, B. (2018). Modelling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm. Computers & Chemical Engineering, 109:9–22.

[38] Reza Sajjadi, S., Hamidi, M., and Hossein Cheraghi, S. (2013). Multi-product capacitated location routing inventory problem. International Journal of Modern Engineering, 13.

[39] Riquelme-Rodriguez, J., Gamache, M., and Langevin, A. (2016). Location arc routing problem with inventory constraints. Computers & Operations Research, 76:84–94.

[40] Seyedhosseini, S., Bozorgi-Amiri, A., and Darei, S. (2014). An integrated location-routing-inventory problem by considering supply disruption. iBusiness, 6:29–37.

[41] Shahin Moghadam, S., Fatemi Ghomi, S., and Karimi, B. (2014). Vehicle routing scheduling problem with cross docking and split deliveries. Computers & Chemical Engineering, 69:98–107.

[42] Singh, T., Arbogast, J., and Neagu, N. (2015). An incremental approach using local-search heuristic for inventory routing problem in industrial gases. Computers & Chemical Engineering, 80:199–210.

[43] Tavakkoli-Moghaddam, R., Forouzanfar, F., and Ebrahimnejad, S. (2013). Incorporating location,

routing and inventory decisions in a bi-objective supply chain design problem with risk pooling. *Journal of Industrial Engineering International*, pages 9–19.

[44] Tsiakis, P., Shah, N., and Pantelides, C. (2001). Design of Multi-echelon Supply Chain Networks under Demand Uncertainty. *Industrial & Engineering Chemistry Research*, 40:3585–3604.

[45] Turkay, M., Oruc, C., Fujita, K., and Asakura, T. (2004). Multi-company collaborative supply chain management with economical and environmental considerations. *Computers & Chemical Engineering*, 28:985–992.

[46] Vicente, J., Relvas, S., and Barbosa-Povoa, A. (2015). Multi-period and Multi-product Inventory Management Model with Lateral Transshipments. In Almeida, J., Oliveira, J., and Pinto, A., editors, *Operational Research. CIM Series in Mathematical Sciences*. Springer.

[47] You, F. and Grossmann, I. (2009). Integrated Multi-Echelon Supply Chain Design with Inventories Under Uncertainty MINLP Models, Computational Strategies. *AIChE Journal*, 56:419–440.

[48] You, F., Wassick, J., and Grossmann, I. (2009). Risk Management for a Global Supply Chain Planning Under Uncertainty Models and Algorithms. *AIChE Journal*, 55:931–946.

[49] Zhalechian, M., Tavakkoli-Moghaddam, R., Zahiri, B., and Mohammadi, M. (2016). Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. *Transportation Research Part E*, 89:182–214.

[50] Zhang, Y., Qi, M., Miao, L., and Liu, E. (2014). Hybrid metaheuristics solutions to inventory location routing problem. *Transportation Research Part E*, 70:305–323.

[51] Zhu, W., Ng, S., Wang, Z., and Zhao, X. (2017). The role of outsourcing management process in improving the effectiveness of logistics outsourcing. *International Journal of Production Economics*, 188:29–40.