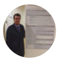
Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem

Article *in* Expert Systems with Applications · April 2020

DOI: 10.1016/j.eswa.2020.113444

3 authors:



Panagiotis Karakostas

University of Macedonia

Angelo Sifaleras

University of Macedonia

Michael C. Georgiadis

Aristotle University of Thessaloniki

Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem

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 introduces the Fleet-size and Mix Pollution Location-Inventory-Routing Problem with Just-in-Time replenishment policy and Capacity Planning. This problem extends the strategic-level decisions of classic LIRP by considering capacity selection decisions and heterogeneous fleet composition. An MIP formulation of this new complex combinatorial optimization problem is proposed and small-sized problem instances are solved using the CPLEX solver. For the solution of more realistic-sized problem instances, a General Variable Neighborhood Search (GVNS)-based framework is adopted. Novel adaptive shaking methods are proposed as intelligent components of the developed GVNS algorithms to further improve their performance. To evaluate the proposed GVNS schemes, several problem instances are randomly generated by following specific instructions from the literature and adopting real vehicles' parameters. Comparisons between these solutions and the corresponding ones achieved by CPLEX are made. The computational results indicate the efficiency of the proposed GVNS-based algorithms, with the best GVNS scheme to produce 7% better solutions than CPLEX for small problems. Finally, the economic and environmental impacts of using either homogeneous or heterogeneous fleet of vehicles are examined.

Keywords: Green Logistics Optimization, Metaheuristics, Location, Inventory, Routing, Fleet Composition

*Corresponding author

Email addresses: pkarakost@cheng.auth.gr (Panagiotis Karakostas), sifalera@uom.gr (Angelo Sifaleras), mgeorg@auth.gr (Michael C. Georgiadis)

Preprint submitted to Expert Systems with Applications *April 6, 2020*

Please cite this paper as:

Karakostas P., Sifaleras A., and Georgiadis M. C., "Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem", Expert Systems with Applications, Vol. 153, Elsevier BV, Article ID 113444, 2020.

The final publication is available at ScienceDirect via https://doi.org/10.1016/j.eswa.2020.113444

1. Introduction

Supply chain management (SCM) traditionally focuses on cost-efficient practices in order to achieve high profit levels. However, the increased socio-environmental concerns have shifted the organizations' focus to a balanced goal which integrates economic, environmental and social goals. These three dimensions are the major pillars of sustainability (Foo et al.,2018). Managing sustainability is characterized by high complexity and based on its strategic role, sustainability affects companies' performance and eventually their growth. Therefore, the efficient SCM is critical for balancing and optimizing sustainability and consequently is the strategic key factor for achieving long-term competitive advantage.

It is commonly accepted that the intertemporal integration of supply chain activities is crucial for any company to achieve competitive advantage (Zhalechian et al.,2016). In this direction, complex supply chain optimization problems, which integrate strategic, tactical and operational decision levels, have been extensively studied in the open literature. The Location-Inventory-Routing problem (LIRP) is one of the most challenging integrated optimization problems (Liu & Lee,2003). It is an NP-hard optimization problem which consists of the Facility Location Problem (FLP), the Inventory Control Problem (ICP) and the Vehicle Routing Problem (VRP) (Javid & Azad,2010). The main objective of this problem is usually the minimization of total cost, consists of location, inventory and routing costs.

Despite the cost efficiency being a prerequisite for sustainable performance, it is not enough for achieving sustainability (Lin et al.,2014a, Lin et al.,2014b). The environmental and consequently the social impacts of supply chain activities must also be improved (Zhang et al.,2016, Xu et al.,2019). Problems which tackle environmental-related decisions are characterized as green optimization problems (Martins & Pato,2019, Bektaş et al.,2019, Skouri et al.,2018, Poonthalir & Nadarajan,2018). The majority of previous works in this area has focused on green routing optimization problems (Yu et al.,2019, Li et al.,2018, Soon et al.,2019). However, as recently noticed by Koç et al. (2016) mentioned that depot- and fleet composition- related decisions also affect emissions. In this direction, several contributions have studied more complex supply chain optimization problems within environmental considerations. Dukkanci et al. (2019) addressed a green location-routing problem. They used a comprehensive modal emission model in order to estimate the emitted pollutants. Zhang et al. (2018) studied a multi-depot emergency location-routing problem with carbon dioxide emissions. Cheng et al. (2017) proposed a green inventory-routing problem with fleet heterogeneity. They highlighted the benefits of using a mixed fleet. Toro et al. (2017) studied the multi-objective green location-routing problem and they highlighted the importance of using more vehicles in shorter routes to minimize both fuel consumption and emissions. Micheli & Mantella (2018) studied an environmentally extended inventory-routing problem with heterogeneous fleet and they examined the effect of different carbon control policies on emissions reduction. Eventhough the environmental-related decisions are critical in achieving sustainability, limited contributions of green LIRP cases have been reported. More specifically, a multi-objective MINLP model for the closed-loop LIRP was proposed (Zhalechian et al.,2016). They used a stochastic-possibilistic approach in order to tackle uncertainty and they developed a hybrid self-adaptive Genetic Algorithm (GA) - Variable Neighborhood Search (VNS) metaheuristic algorithm to solve large-sized instances. Karakostas et al. (2019b) proposed a Pollution LIRP and they illustrated the applicability of Basic VNS (BVNS) metaheuristic algorithms on medium-sized problem instances. Table 1 summarizes the main LIRP contributions.

Table 1: Key literature contributions on LIRP

Reference	P.T.[1]	D.T.[2]	C.T.[3]	R.P.[4]	F.C.[5]	E.D.[6]	C.P.[7]	Model	S.M.[8]
(Javid & Azad, 2010)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	x	MINLP	Metaheuristic
(Tavakkoli-Moghaddam et al., 2013)	Single	Stochastic	Single	(Q,R)	Heterogeneous	x	x	MINLP	Exact
(Guerrero et al., 2013)	Multiple	Deterministic/Variable	Single	Order up to level	Homogeneous	x	x	MIP	Metaheuristic
(Seyedhosseini et al., 2014)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	x	MINLP	Exact/Metaheuristic
(Chen et al., 2014)	Single	Fuzzy	Single	(T,R) Periodic	Homogeneous	x	x	MIP	Metaheuristic
(Nekooghadirli et al., 2014)	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	x	x	MINLP	Metaheuristic
(Zhang et al., 2014)	Multiple	Deterministic/Variable	Single	Flexible	Homogeneous	x	x	MIP	Metaheuristic
(Liu et al., 2015)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	x	MINLP	Metaheuristic
(Zhalechian et al., 2016)	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	x	x	MMINLP	Metaheuristic
(Ghorbani & Akbari Jokar, 2016)	Multiple	Deterministic/Variable	Multiple	(Q,R)	Homogeneous	x	x	MIP	Metaheuristic
(Hiassat et al., 2017)	Multiple	Deterministic/Variable	Single*	-	Homogeneous	x	x	MIP	Metaheuristic
(Rayat et al., 2017)	Multiple	Stochastic	Multiple	Continuous Review	Heterogeneous	x	✓	BOMIP	Metaheuristic
(Vahdani et al., 2018)	Multiple	Deterministic	Multiple	-	Heterogeneous	x	✓	MOMIP	Metaheuristic
(Rafie-Majd et al., 2018)	Multiple	Stochastic	Multiple	(Q,R)	Heterogeneous	x	x	MINLP	Heuristic
(Asadi et al., 2018)	Single	Stochastic	Sinlge	(S-1,S)	Homogeneous	✓	x	BOMINLP	Metaheuristic
(Habibi et al., 2018)	Single	Stochastic	Single	(S-1,S)	Homogeneous	x	x	MINLP	Metaheuristic
(Tavana et al., 2018)	Multiple	Deterministic	Multiple	Order-based	Heterogeneous	x	x	MOMILP	Metaheuristic
(Saragih et al., 2019)	Single	Stochastic	Single	(Q,R)	Homogeneous	x	✓	MINLP	Metaheuristic
(Saif-Eddine et al., 2019)	Multiple	Deterministic	Single	-	Homogeneous	x	x	MIP	Metaheuristic
(Zheng et al., 2019)	Single	Stochastic	Single	(T,s)	Homogeneous	x	x	MINLP	Exact
This work	Multiple	Deterministic/Variable	Single	Just-in-Time	Heterogeneous	✓	✓	MIP	Metaheuristic

[1] *Period Type*, [2] *Demand Type*, [3] *Commodity Type*, [4] *Replenishment Policy*, [5] *Fleet Composition*, [6] *Environmental Decisions*, [7] *Capacity Planning*, [8] *Solution Method*

The importance of selecting an appropriate replenishment policy in the green IRP problem has been clearly highlighted in previous studies. The Just-in-Time (JiT) replenishment policy is a popular inventory management strategy based on the lean management philosophy and the increased customer satisfaction. Recent studies have shown that this policy positively affects the sustainable performance of a company (Kong et al.,2018, Wang & Ye,2018). This is due to the elimination of storage activities and consequently the relative waste. Moreover, the significance of facilities-related decisions on a company's sustainable performance is indisputable. That being said, the capacity planning of facilities is also critical for achieving sustainability due to its strategic nature (Aldis,2017).

This work addresses a new variant of the LIRP, the Fleet-size and Mix Pollution LIRP with JiT replenishment policy and capacity planning (FSMPLIRP). This new NP-hard problem considers further strategic level decisions, such as the capacity planning and fleet composition. The JiT replenishment policy is the only appropriate in some emergency supply chain networks, such as the medical supply chains. However, it reduces the flexibility on route scheduling and makes even harder the effort of building efficient routes. For the efficient solution of the underlying problem, the development of problem-specific solution methods is crucial. The capability of an algorithm to use past experience in order to improve its performance is a key feature of intelligent optimization. Therefore, in this work we develop General Variable Neighborhood Search (GVNS) metaheuristic algorithms within adaptive shaking mechanisms in an effort to improve their performance and solve a supply chain problem of significant industrial interest. The proposed modeling framework and solution approaches can provide the basis for the development of an expert system that can assist decision makers to derive rigorous and fast decisions related to the operation and design of such supply chains. The main contributions of this work are summarized as follows:

- An MIP formulation for the Fleet-size and Mix Pollution Location Inventory Routing Problem .

- New adaptive shaking methods, as intelligent components in the developed GVNS algorithms for the solution of the above problem.

- Investigation of different variants of the solution approaches.

- Development of a self-contained solver for the problem under consideration.

- Useful managerial insights are provided by solving a number of problem instances.

The structure of this work is as follows. In Section 2 the problem statement is given while Section 3 provides a detailed presentation of the proposed solution algorithms. Section 4 presents several computational analyses and summarizes key managerial insights. Finally, conclusions are drawn in Section 5.

2. Problem statement

The FSMPLIRP is defined as a complete graph $G = (V, E)$, where V denotes the set of nodes including both the set of customers $I = \{1, ..., N_{Customers}\}$ and the set of potential depots $J = \{1, ..., N_{Depots}\}$ and $E = \{(v, v_1) : v, v_1 \in V, v \neq v_1\}$ is the set of edges. Each customer has a period-dependent demand for a single-type of product and it is served by a heterogeneous fleet of vehicles. A vehicle has a fixed usage cost and a specific capacity level. A mixed integer programming (MIP) model is proposed to describe this problem. The model is an extension of a recently proposed PLIRP formulation (Karakostas et al.,2019b) and it tackles a more complex variant of the LIRP by considering facility capacity planning, fleet composition and JiT replenishment policy. The use of JiT replenishment policy means that the delivered and the demanded quantity of product must be equal for each customer in each time period. Thus, the formation of efficient routes gets even harder than the case of PLIRP. For clarity reason the model sets, parameters and variables are provided in Tables 2, 3, 4 and 5.

Table 2: 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
L	set of capacity levels

Table 3: Vehicles' parameters.

Parameter	**Explanation**	**Value** (Cheng et al.,2017;Koç et al.,2014;Karakostas et al.,2019b)
ϵ	fuel-to-air mass ratio	1
g	gravitational constant (m/s^2)	9.81
ρ	air density (kg/m^3)	1.2041
CR	coefficient of rolling resistance	0.01
η	efficiency parameter for diesel engines	0.45
f_c	unit fuel cost ($Euros/L$)	0.7382
f_e	unit CO_2 emission cost ($Euros/kg$)	0.2793
f_d	driver wage ($Euros/s$)	0.0025
σ	CO_2 emitted by unit fuel consumption (kg/L)	2.669
$HVDF$	heating value of a typical diesel fuel (kj/g)	44
ψ	conversion factor (g/s to L/s)	737
θ	road angle	0
τ	acceleration (m/s^2)	0
CW_k	curb weight (kg)	3500
EFF_k	engine friction factor ($kj/rev/L$)	0.25
ES_k	engine speed (rev/s)	39
ED_k	engine displacement (L)	2.77
CAD_k	coefficient of aerodynamics drag	0.6
FSA_k	frontal surface area (m^2)	9
$VDTE_k$	vehicle drive train efficiency	0.4
Q_k	loading capacity of vehicle k	instance-dependent
VFC_k	usage cost of vehicle k	1200 or 1400

The value of parameters f_e and f_d are converted into Euro currency (26th of February, 2018). The usage cost for light-duty vehicles taken as 1200 Euros and for the case of medium-duty vehicles is 1400 Euros.

Table 4: Non-vehicle related FSMPLIRP model parameters.

Notation	**Explanation**
f_{jl}	fixed opening cost of depot j with capacity level l
C_{jl}	storage capacity of depot j with capacity level l
h_i	unit inventory holding cost of customer i
d_{it}	period-variable demand of customer i
c_{ij}	travelling cost of locations pair (i, j)
s_r	the value of the speed level r

Table 5: GLIRP model variables.

Notation	**Explanation**
y_{jl}	1 if depot j with capacity level l is opened; 0 otherwise
z_{ij}	1 if customer i is assigned to depot j; 0 otherwise
vs_{kt}	1 if vehicle k is selected in period t; 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
a_{vikt}	load weight by travelling from node v to the customer i with vehicle k in period t
$zz_{v_1v_2ktr}$	1 if vehicle k travels from node $v1$ to $v2$ in period t with speed level r

In this work, a comprehensive fuel consumption model from the literature is adopted (Cheng et al.,2017). To reduce the length of some parts of the objective function, the next formulas are provided.

- $\alpha = \tau + gCR \sin\theta + gCR \cos\theta$
- $\gamma_k = \frac{1}{1000VDTE_k\eta}$
- $\beta_k = 0.5CAD\rho FSA_k$
- $\lambda = \frac{HVDF}{\psi}$

$$
\begin{aligned}
&\min \sum_{j\in J} f_{jl}y_{jl} + \sum_{i\in I} h_i \sum_{t\in H} \tfrac{1}{2}d_{it} \quad + \sum_{i\in V}\sum_{j\in V}\sum_{t\in H}\sum_{k\in K} c_{ij}x_{ijkt} \\
&+ \sum_{i\in V}\sum_{j\in V}\sum_{k\in K}\sum_{t\in H} \Bigg\{ \lambda\left(f_c + (f_e\sigma)\right) \Bigg(\sum_{r\in R} \frac{\left(zz_{ijktr}\ EFF_k\ ES_k\ ED_k\ c_{ij}\right)}{s_r} \\
&+ \Bigg(\alpha\gamma_k \left(CW_k\ x_{ijkt} + a_{ijkt}\right)\ c_{ij} \Bigg) + \Bigg(\beta_k\ \gamma_k \sum_{r\in R} \left(s_r\ zz_{ijktr}\right)^2 \Bigg) \Bigg) \Bigg\} \\
&+ \sum_{i\in V}\sum_{j\in V}\sum_{k\in K}\sum_{t\in T}\sum_{r\in R} f_d \frac{\left(zz_{ijktr}\ c_{ij}\right)}{s_r} + \textstyle\sum_{k\in K}\sum_{t\in H} vs_{kt}VFC_k
\end{aligned}
\tag{1}
$$

Subject to

$$
vs_{kt} \leq \sum_{v\in V}\sum_{v_1\in V} x_{vv_1kt}, \quad \forall k\in K, \forall t\in H, v\neq v_1 \tag{2}
$$

$$
x_{vv_1kt} \leq vs_{kt} \quad \forall v, v_1 \in V, v\neq v_1, \forall k\in K, \forall t\in H \tag{3}
$$

$$
\sum_{r\in R} zz_{ijktr} = 1 \quad \forall i,j\in V, \forall k\in K, \forall t\in H \tag{4}
$$

$$
\sum_{i\in V} a_{ijkt} - \sum_{i\in V} a_{jikt} = q_{jkt}PW \quad \forall j\in I, \forall k\in K, \forall t\in H \tag{5}
$$

$$
\sum_{j\in V} x_{ijkt} - \sum_{j\in V} x_{jikt} = 0 \qquad\qquad \forall i\in V, \forall k\in K,\ \forall t\in H \tag{6}
$$

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

$$
\sum_{j\in V}\sum_{k\in K} x_{jikt} \leq 1 \quad \forall t\in H,\ \forall i\in I \tag{8}
$$

$$
\sum_{i\in I}\sum_{j\in J} x_{ijkt} \leq 1 \quad \forall k\in K,\ \forall t\in H \tag{9}
$$

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

$$\sum_{i \in I} q_{ikt} \leq Q_k \quad \forall k \in K,\ \forall t \in H \tag{11}$$

$$\sum_{j \in J} z_{ij} = 1 \quad \forall i \in I \tag{12}$$

$$z_{ij} \leq y_{jl} \quad \forall i \in I,\ \forall j \in J \tag{13}$$

$$\sum_{i \in I} \left(z_{ij} \sum_{t \in H} d_{it} \right) \leq C_{jl} \quad \forall j \in J, \forall l \in L \tag{14}$$

$$\sum_{u \in I} x_{ujkt} + \sum_{u \in V \setminus \{i\}} x_{iukt} \leq 1 + z_{ij} \quad \forall i \in I,\ \forall j \in J,\ \forall \mathrm{k} \in \mathrm{K}, \forall t \in H \tag{15}$$

$$\sum_{i \in I} \sum_{k \in K} \sum_{t \in H} x_{jikt} \geq y_{jl} \quad \forall j \in J, \forall l \in L \tag{16}$$

$$\sum_{i \in I} x_{jikt} \leq y_{jl} \quad \forall j \in J, \forall l \in L, \forall k \in K,\ \forall t \in H \tag{17}$$

$$\sum_{k \in K} q_{ikt} = d_{it}, \quad \forall i \in I, \forall t \in H \tag{18}$$

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

$$\sum_{j \in V} x_{ijkt} \leq M q_{ikt} \quad \forall i \in I,\ \forall t \in H,\ \forall \mathrm{k} \in \mathrm{K} \tag{20}$$

$$\mathrm{x}_{ijkt} \in \{0, 1\} \quad \forall i \in I,\ \forall j \in J,\ \forall t \in H,\ \forall k \in K \tag{21}$$

$$y_{jl} \in \{0, 1\} \quad \forall j \in J, \forall l \in L \tag{22}$$

$$z_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J \tag{23}$$

$$q_{ikt} \leq \min\{Q_k, d_{it}\} \quad \forall i \in I,\ \forall k \in K\ ,\forall t \in H \tag{24}$$

The objective criterion of this model is the minimization of the total cost which consists of the facilities' opening costs, the average inventory holding costs, general routing costs, fuel and CO_2 emissions costs, driver wages and vehicle usage costs. Constraints (2) ensure that a vehicle is selected in a period only if a route has been scheduled for it in that period. Constraints (3) guarantee that a vehicle will move through a pair of nodes in a period, only if it is selected in that period. Constraints (4) impose the selection of a specific speed level for traveling through two nodes in each time period. Constraints (5) satisfy the product flow balance and simultaneously act as subtour elimination constraints. Constraints (6) guarantee the equilibrium between the interior and exterior flow of vehicles. A customer will be serviced by one vehicle at most in each time period, as it is imposed by Constraints (7) and (8). Constraints (9) force a vehicle to not perform more than one route per time period. Constraints (10) ensure that a vehicle will not move through two depot locations. The product quantity delivered with a vehicle must not exceed its capacity, as it is imposed by Constraints (11). According to Constraints (12) a vehicle will move from a depot to a customer, only if that customer is assigned to the depot. A customer can be assigned only to an open depot based on Constraints (13). Constraints (14) ensure the observance of depots' capacities. Constraints (15) impose a customer to connect with a depot, only if that customer is allocated to that depot. A vehicle departures from a depot only if that depot is opened according to Constraints (16) and (17). A customer is visited at a specific period, only if a replenishment is scheduled for that period, according to Constraints (19). The last four set of constraints declare the nature of the decision variables.

3. General variable neighborhood search-based heuristics

Variable neighborhood search (VNS) is a metaheuristic framework known for its flexibility and simplicity on building efficient heuristic algorithms (Hansen et al.,2010). A VNS algorithm mainly consists of the three following components:

- An improvement phase,
- a shaking phase,
- a neighborhood change step (Hansen et al.,2017).

In the improvement phase, local search operators are systematically applied in order to improve a given solution, while the shaking phase aims at resolving local optimum traps . The neighborhood change step leads the exploration of the solution space (Hansen et al.,2017). Several contributions in the recent literature have applied the VNS framework to solve efficiently hard optimization problems (Fuqing et al.,2019, Xu & Cai, 2018, Simeonova et al.,2018).

Variable neighborhood descent (VND) is a VNS variant in which a number of local search operators are applied iteratively with respect to an adopted neighborhood change strategy (Hansen et al.,2017). According to the neighborhood change strategy, the following sequential VND schemes are formed:

- **Basic VND (bVND)**. Each time an improved solution is found, the search continues with the first operator (Hansen et al.,2017).
- **Pipe VND (pVND)**. If an improved solution is found within an operator, the search continues with that operator (Hansen et al.,2017).
- **Cyclic VND (cVND)**. The search continues with the next operator regardless the improvements (Hansen et al.,2017).
- **Union VND (uVND)**. It is also known as Multiple neighborhood search. The search is applied in the union of all neighborhood structures (Hansen et al.,2017).

- **Extended VND (eVND)**. This VND variant extends bVND by specifying a parameter (m) which indicates the improvement depth. More specifically, the search switches to the first operator either when m improvements are achieved by the current operator or exactly one improvement is made within the current operator (Lai & Hao,2016).

- **Adaptive VND (aVND)**. This variant uses one of the previous VND schemes but in each iteration the order of the neighborhoods is changed mainly according to their success in the previous iteration (Todosijević et al.,2016).

General variable neighborhood search (GVNS) is a widely used VNS variant, which uses a VND method as its main improvement phase (Hansen et al.,2017). Recently many GVNS schemes have been efficiently applied on solving hard combinatorial optimization problems (De Armas et al.,2015, Bezerra et al., 2018, Karakostas et al.,2019a, Mikić et al.,2019).

3.1. Initial solution

To build an initial feasible solution, a three-phase construction method is proposed. Location, capacity planning and allocation decisions are made in its first phase. In an effort to find the minimum required number of depots, a ratio-based depots' selection method is applied, as presented in our previous work (Karakostas et al.,2019a). If more than one depots are needed for servicing the given customers, then a nearest customer allocation procedure is used for each opened depot. In the next phase, the deliveries are set equal to their corresponding demands for each customer in each time period and the routes are built by applying a modified Nearest Neighbor heuristic (Flood,1956). Finally, the speed levels for traveling through the links of the designed network are randomly set.

3.2. Local search operators & pVND

This section describes eight local search operators which are designed to explore the solutions of the corresponding neighborhood structures. These operators are the following:

Inter-route Relocate (N_1). This operator selects two customers assigned to different routes. Then, it removes the first selected customer from its current position and relocates it to the next position of the second selected customer. The initial routes of the two selected customers can either be assigned to the same depot or different depots.

Opened-Closed Depots Exchange (N_2). In this operator for each closed depot the maximum capacity level is selected and examined if that depot can replace one of the currently opened depots. It is mainly examined if the capacity of the closed depot is enough to deal with the total demand of customers allocated to the, potentially to be exchanged, opened depot.

Intra-route Relocate (N_3). This operator selects two customers allocated to the same depot and moves the first selected customer from its current position to the next position of the second selected customer.

Inter-route Exchange (N_4). This operator swaps two selected customers which they are assigned to different routes. Similarly to the Inter-route Relocate, the routes can be allocated to the same depot or not.

Intra-route 2-Opt (N_5). It selects two pairs of successive customers, assigned to the same route, (i, j) and (k, l). Next, it breaks them and reconnects them differently, such as (i, k) and (j, l).

One Medium-Two Light Vehicles Exchange (N_6). This operator selects two currently used light-duty vehicles and examines if the serviced, by those vehicles, customers can be serviced by one unselected medium-duty vehicle.

Select Depot Capacity Level (N_7). In this operator the most cost-efficient capacity level is selected for each opened depot with respect to the total demand of its customers.

Medium-To-Light Vehicles Exchange (N_8). This operator selects a used medium-duty vehicle and examines if the total demand of its customers can be serviced by a light-duty vehicle, in order to perform an exchange between those two vehicles.

These local search operators are included in two pVND methods. The first method contains operators $N_1 - N_5$, while the second one contains operators $N_1 - N_6$. The pVND is selected due to its efficiency in solving hard optimization problems, as it is highlighted in our previous work (Karakostas et al.,2019a). An adaptive search strategy is also adopted (Best improvement is applied on small- and medium-sized problem instances, while first improvement is used for the case of large-sized instances). The pseudocodes of the proposed pVND schemes are summarized in Algorithms 1 and 2.

Algorithm 1 pipe-VND 1

```
procedure pVND_1(S, l_max)
    l = 1
    while l ≤ l_max do
        select case(l)
        case(1)
        S' ← N_1(S)
        case(2)
        S' ← N_2(S)
        case(3)
        S' ← N_3(S)
        case(4)
        S' ← N_4(S)
        case(5)
        S' ← N_5(S)
        end select
        if f(S') < f(S) then
            S ← S'
        else
            l = l + 1
        end if
    end while
    Return S
end procedure
```

Algorithm 2 pipe-VND 2

```
procedure pVND_2(S, l_max)
    l = 1
    while l ≤ l_max do
        select case(l)
        case(1)
        S' ← N_1(S)
        case(2)
        S' ← N_2(S)
        case(3)
        S' ← N_3(S)
        case(4)
        S' ← N_4(S)
        case(5)
        S' ← N_5(S)
        case(6)
        S' ← N_6(S)
        end select
        if f(S') < f(S) then
            S ← S'
        else
            l = l + 1
        end if
    end while
    Return S
end procedure
```

Operators N_7 and N_8 are applied within the pVND methods as an integrated improvement phase.

3.3. Shaking procedures

Diversification methods are critical components of metaheuristic algorithms (Xu & Cai,2018). They are strategies for escaping from local optimum solutions by using properly modified local search operators. In this work, five shaking operators are designed:

- **Inter-route Exchange Shaking (S_1).** It works as the local search operator N_4 with the difference that the two customers are selected randomly.
- **Opened-Closed Depots Exchange (S_2).** This shaking operator functions similar to N_2. The main difference is that the closed depot is selected randomly.
- **Intra-route Relocate (S_3).** In this operator two customers are randomly selected in each time period. Then, this shaking operator performs like as N_3.
- **Select Depot Capacity Level Shaking (S_4).** This operator selects randomly an opened depot and changes the capacity level of that depot, with respect to the total demand of the customers serviced by it.
- **Light2Medium Vehicles Exchange Shaking (S_5).** Initially, a time period is randomly selected and then a selected light-duty vehicle is exchanged with a medium-duty vehicle.

The above operators are embedded in two shaking procedures (the first does not include the S_5). Their pseudocodes are provided in Algorithms 3 and 4.

Algorithm 3 Shaking procedure 1

```
procedure SHAKE_1(S, l)
    select case(l)
    case(1)
    S' ← S_1(S)
    case(2)
    S' ← S_2(S)
    case(3)
    S' ← S_3(S)
    case(4)
    S' ← S_4(S)
    end select
      Return S'
```

Algorithm 4 Shaking procedure 2

```
procedure SHAKE_2(S, l)
    select case(l)
    case(1)
    S' ← S_1(S)
    case(2)
    S' ← S_2(S)
    case(3)
    S' ← S_3(S)
    case(4)
    S' ← S_4(S)
    case(5)
    S' ← S_5(S)
    end select
      Return S'
```

The most commonly used diversification method within VNS is the intensified shaking, which randomly selects a shaking operator and applies it k times, where k denotes the intense of diversification and it is $1 \leq k \leq k_{max}$, with k_{max} being the shaking strength. Additional to the intensified shaking, this work proposes two adaptive shaking procedures. Initially, the five shaking operators are ordered in a set. According to that initial order, two adaptive shaking procedures are formed. In the first procedure the initial order of operators is based on their computational complexity, while in the second one their ordering is performed randomly. However, both of them are executed similarly. More specifically, in each GVNS iteration and for a specific k value, the shaking operators are executed sequentially (shaking operator - pVND - solution renewal check). A five positions array is used to count the improvements, achieved by using each shaking operator. Each position is matched with one shaking operator and in case of finding a new best solution, the value in this position is increased by one. In the next iteration of GVNS, the sequence of shaking operators is re-ordered according to the number of improvements recorded in the previous iteration. If no improvements or the same number of improvements are achieved during an iteration, the initial order is adopted for the next iteration. Essentially, the core difference between the adaptive shaking schemes and the intensified shaking lies in the manner the shaking operators are handled.

Focused on the adaptive shaking strategies, a reduced scheme is also examined. In particular, in each GVNS iteration, different shaking operators are applied for different k values. For instance, for $k = 1$, the first shaking operator is applied, for $k = 2$, the next operator and so on. If all operators are applied and variable k has not reached the k_{max} value, the diversification process will continue from the first operator. In each next GVNS iteration, the re-ordering step is applied such as in the previously discussed adaptive shaking strategies.

3.4. GVNS schemes

The use of different components leads to different GVNS schemes. Moreover, the structure of numerical analyses may impose the formation of further GVNS schemes. From a problem solution perspective, two cases of GVNS schemes are met:

- Case_1: GVNS schemes for solving the homogeneous case of the problem.
- Case_2: GVNS schemes for solving the heterogeneous case of the problem.

From a shaking strategy perspective, three cases of GVNS schemes are investigated:

- Case_1: GVNS schemes which use the intensified shaking.
- Case_2: GVNS schemes which use the adaptive shaking method with complexity-based initialization.
- Case_3: GVNS schemes that they use the adaptive shaking method with random initial order.

Therefore, the following main GVNS are defined:

- $GVNS\_1$: This heuristic is proposed for solving the homogeneous case of the problem and uses the intensified shaking as its diversification strategy.
- $GVNS\_2$: This GVNS scheme solves the same problem case as the $GVNS\_1$, but it uses the adaptive shaking with complexity-based initialization.
- $GVNS\_3$: An other heuristic for solving the homogeneous case of the problem which uses the adaptive shaking with random initial order.
- $GVNS\_4$: This GVNS scheme is proposed for solving the heterogeneous case of the problem. The intensified shaking is used.
- $GVNS\_5$: This heuristic solves the heterogeneous case of the problem and uses the adaptive shaking with complexity-based initialization.
- $GVNS\_6$: This GVNS heuristic solves the heterogeneous case of the problem and the adaptive shaking with random-based initialization is used.

The pseudocodes of the first three GVNS schemes are provided in Algorithms 6, 7 and 8. However, before the presentation of these pseudocodes, the re-ordering mechanism of shaking operators is provided in Algorithm 5. The *ShakingOrder* is the ordered set of shaking operators, *InitialOrder* keeps the initial order of the shaking operators and *ShakingOperatorsChecked* is a logical array which indicates if a shaking operator is selected during the re-ordering phase.

Algorithm 5 Re-ordering mechanism

```
procedure ADAPTIVE_ORDER(ShakingOrder, InitialOrder)
    if no improvement is found in any neighborhood then
        ShakingOrder = InitialOrder
    end if
    if an improvement is found then
        for i ← 1, 5 do
            l = Operator with maximum number of improvements
            ShakingOperatorChecked(l) = .true.
            ShakingOrder(i) = l
        end for
    end if
    return ShakingOrder
end procedure=0
```

Algorithm 6 General VNS 1

```
procedure GVNS_1(S, k_max, max_time, l_max)
    while time ≤ max_time do
        for k ← 1, k_max do
            S* = Shake_1(S, l)
            S' = pVND_1(S*, l_max)
            S* = N_7(S')
            if f(S*) < f(S) then
                S ← S*
            end if
        end for
    end while
    return S
```

Algorithm 7 General VNS 2

```
procedure GVNS_2(S, k_max, max_time, l_max)
    while time ≤ max_time do
        ShakingOrder = Adaptive_Order(ShakingOrder, InitialOrder)
        for k ← 1, k_max do
            for i ← 1, 5 do
                l = ShakingOrder(i)
                S* = Shake_1(S, l)
                S' = pVND_1(S*, l_max)
                S* = N_7(S')
                if f(S*) < f(S) then
                    S ← S*
                end if
            end for
        end for
    end while
    return S
```

Algorithm 8 General VNS 3

```
procedure GVNS_3(S, k_max, max_time, l_max)
    for i ← 1, 5 do
        InitialOrder(i) = i
    end for
    ShakingOrder = Shuffle(InitialOrder)
    while time ≤ max_time do
        ShakingOrder = Adaptive_Order(ShakingOrder, InitialOrder)
        for k ← 1, k_max do
            for i ← 1, 5 do
                l = ShakingOrder(i)
                S* = Shake_1(S, l)
                S' = pVND_1(S*, l_max)
                S* = N_7(S')
                if f(S*) < f(S) then
                    S ← S*
                end if
            end for
        end for
    end while
    return S
```

The pseudocodes of $GVNS\_4$, $GVNS\_5$ and $GVNS\_6$ are omitted, as they are similar to the previously provided GVNS schemes. Their differences are the use of $pVND\_2$ and operator N_8, which is executed exactly after operator N_7. More specifically, to solve the heterogeneous problem case efficiently, further local search and shaking operators are required. These operators perform proper changes in order to improve the fleet composition.

Due to the fact that the reduced adaptive shaking strategy is a special case of the adaptive shaking strategy, each GVNS scheme, which uses this shaking approach, is defined as $GVNS\_X_R$, where $GVNS\_X$ is the corresponding GVNS scheme with no-reduced adaptive shaking. For instance, the reduced variant of $GVNS\_3$ is the $GVNS\_3_R$ and its pseudocode is provided in Algorithm 9 .

Algorithm 9 General VNS 3 with reduced adaptive shaking

```
procedure GVNS_3_R(S, K_max, max_time, l_max)
    for i ← 1, 5 do
        InitialOrder(i) = i
    end for
    ShakingOrder = Shuffle(InitialOrder)
    while time ≤ max_time do
        ShakingOrder = Adaptive_Order(ShakingOrder, InitialOrder)
        i = 1
        for k ← 1, k_max do
            l = ShakingOrder(i)
            S* = Shake_1(S, l)
            S' = pVND_1(S*, l_max)
            S* = N_7(S')
            if f(S*) < f(S) then
                S ← S*
            end if
            i = i + 1
            if i > 5 then
                i = 1
            end if
        end for
    end while
    return S
```

The results of computational experiments on the heterogeneous case of the problem, show a potential benefit with increasing fleet diversity (for further details see Subsection 4.3). Those GVNS schemes use the $Shake\_2$ instead of $Shake\_1$.

Finally, it should be mentioned that several auxiliary methods have been developed to guarantee the feasibility of the obtained solutions. For instance, a method which examines the existence of sub-routes in a selected route.

4. Computational analysis and results

4.1. Computing environment

The MIP formulation of the studied problem was implemented in GAMS (GAMS 24.9.1) (Brooke et al.,1998) and its instances were solved by CPLEX 12.7.1.0 solver. The time limit for solving small-sized instances was set at two hours, while for the medium- and large-sized instances the time limit was increased to up to three hours. The proposed algorithms were coded in Fortran and they were executed by Intel Fortran compiler 18.0 using the optimization option /O3. Both CPLEX and Intel Fortran compiler ran on a laptop PC running Windows 10 Home 64-bit with an Intel Core i7-6700 CPU at 2.6 GHz and 16 GB RAM. The execution time limit for the designed heuristic algorithms was set at $60s$.

4.2. Problem instances

Due to the fact that the FSMPLIRP is introduced in this work, there is no available test instances in the literature. Thus, 30 new problem instances were randomly generated, using the instructions given by Zhang et al. (2014). The vehicle fixed cost of light-duty vehicles is randomly generated with a Normal distribution with parameters $\mu = 1000$ and $\sigma = 500$, while the cost of medium-duty vehicles is calculated as $floor((light\_cost + (light\_cost * (20\% + rand(0.5, 5)))))$. Each problem instance has a name formed as X-Y-Z, where X denotes the number of potential depots, Y the number of customers and Z the number of time periods. The set of generated problem instances are available in `http://pse.cheng.auth.gr/index.php/publications/benchmarks/`.

4.3. Parameter setting & computational results

Before the presentation of the experimental study, an overview of the proposed solution method is provided in Figure 1.

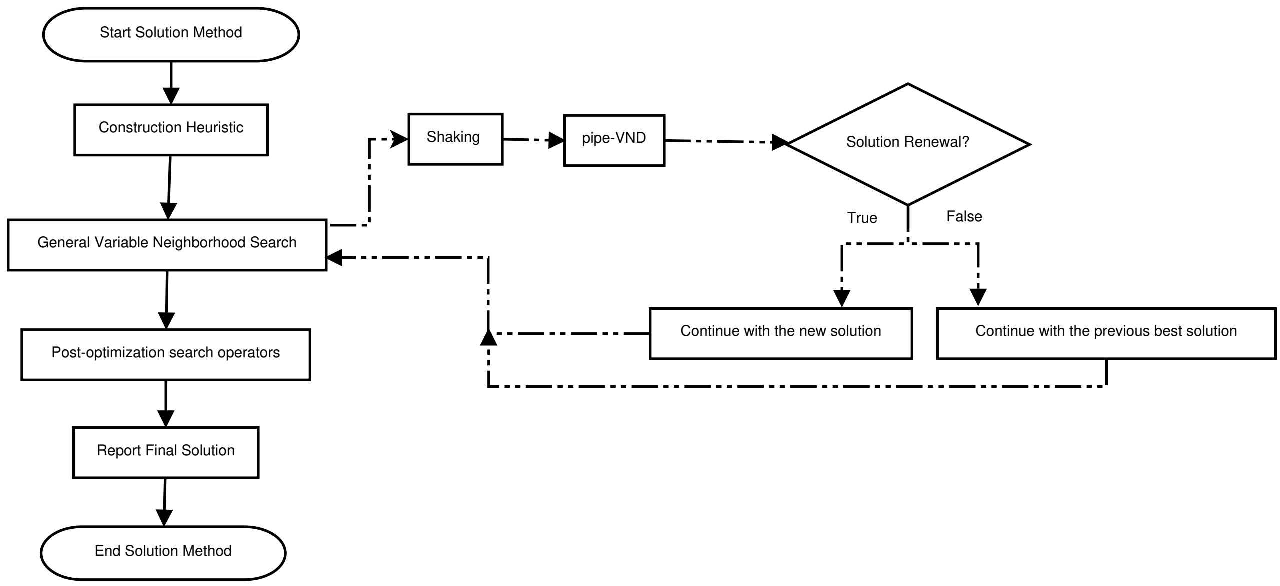


Figure 1: Flowchart of the proposed solution method.

A critical parameter of a VNS-based heuristic algorithm is k_{max}. In this regard, a parameter estimation is performed in order to select the most efficient value of this parameter. The examined values of k_{max} are 10, 12, 15, 20 and 25. For this estimation process, the $GVNS\_1$ is used (light-duty vehicles case). Table 6 summarizes the total cost achieved for each problem instance and different values of k_{max}. It should be mentioned that in all presented results, the reported value of each instance is the average solution of 10 runs.

Table 6: k_{max} analysis on the $GVNS\_1$ performance on 30 GLIRP

Instance	$k_{max} = 10$	$k_{max} = 12$	$k_{max} = 15$	$k_{max} = 20$	$k_{max} = 25$
4-9-3	19950.64	19950.64	20025.74	19965.42	20001.45
4-10-3	20776.81	20401.16	20323.12	20442.53	20484.32
4-10-5	16890	16757.12	16639.51	16994.84	16773.05
4-12-5	20745.54	21537.64	19257.65	19241.52	19298.63
4-15-3	10205.15	10202.94	10202.69	10202.63	10207.29
5-12-3	12966.13	12966	12966.47	12966.83	12982.32
5-15-3	15980.28	15973.33	15979.88	15970.49	15979.27
5-15-5	21963.65	22156.5	21973	21984.31	22028.07
5-18-3	23382.51	22843.35	22989.54	23503.2	22795.83
5-20-3	19082.94	19080.76	19145.74	19095.92	19083.83
6-40-5	22053.01	22116.36	22086.75	22166.08	22051.46
7-52-5	16565.59	16459.8	16475.07	16449.83	16602.87
7-55-7	20640.6	20740.16	20680.7	20734.96	20680.7
8-60-5	25158.09	25270.78	24917.71	25192.96	25094.77
8-65-7	45432.84	46333.11	46389.98	46813.38	46404.39
9-70-5	27257.63	27257.63	27257.63	26954.93	26422.93
9-75-7	29229.05	29235.23	29256.86	29229.98	29234.61
9-85-5	23312.07	23113.37	23355.22	23346.96	23307.15
9-88-7	28413.28	28298.62	28497.24	28622.26	28606.53
10-90-7	25664.05	25744.83	25744.83	25651.54	25744.6
15-100-7	21079.91	21175.92	20676.81	21168.42	21061.69
15-100-10	32776.21	33164.21	32454.69	33162.97	33162.97
15-120-10	32001.47	31998.65	31712.23	31869.61	31866.53
20-150-10	27251.23	27247.53	27011.78	27251.23	27242.03
20-180-12	56623.45	56001.89	55474.01	56363.05	56779.96
25-200-12	53858.84	55481.55	53660.57	55448.09	55502.41
30-250-10	40514.82	40608.62	40608.62	40621.34	40339.15
30-270-10	40604.43	40001.64	39793.64	39817.2	39804.99
35-300-10	69917.79	71524.05	70530.91	70429.36	70638.92
35-310-12	70241.78	69334.98	70366.98	69721.21	70114.25
Average	29684.66	29765.95	**29350.96**	29022.74	29676.57

In accordance with the average values of the previously reported results, it is obvious that $k_{max} = 15$ produces slightly better solutions than the other tested values. This minor improvement is mainly based on the results achieved on ten small-sized and ten large-sized instances. The selected strength of shaking presumably permits more iterations of the improvement phase than the more intense shaking options and better exploration than the limited k_{max} choices.

To fairly compare the intensified shaking with the two proposed adaptive shaking methods (actually their corresponding GVNS schemes), the same k_{max} value is also used in the adaptive cases. Table 7 provides the average and best results obtained by $GVNS\_1$, $GVNS\_2$ and $GVNS\_3$.

Table 7: Average and best values of $GVNS\_1$, $GVNS\_2$ and $GVNS\_3$

Instance	$GVNS_1\_Avg$	$GVNS_1\_Best$	$GVNS_2\_Avg$	$GVNS_2\_Best$	$GVNS_3\_Avg$	$GVNS_3\_Best$
4-9-3	20025.74	19893.35	19965.42	19893.35	19965.42	19893.35
4-10-3	20323.12	20211.22	20388.18	20386.88	20477.19	20236.63
4-10-5	16639.51	16639.47	16654.57	16639.49	16668.95	16639.52
4-12-5	19257.65	19218.74	19330	19218.7	19999.71	19232.61
4-15-3	10202.69	10197.83	10204.27	10199.25	10206.14	10197.82
5-12-3	12966.47	12965.52	12977.22	12965.53	12966.05	12965.53
5-15-3	15979.88	15968.03	15978.63	15967.97	15982.42	15968.01
5-15-5	22040.32	21811.93	21973.37	21829.92	22097.31	22061.04
5-18-3	22989.54	22034.8	22393.16	22044.37	22769.88	22048.34
5-20-3	19145.74	19072.08	19097.41	18970.59	19109.76	18969.02
6-40-5	22086.75	21869.92	22033.75	21955.77	22113.73	21864.69
7-52-5	16475.07	16346.4	16492.06	16338.96	16523.26	16357.35
7-55-7	20680.7	20483.23	20289.07	20133.6	20617.13	20220.49
8-60-5	25209.82	24851.46	24917.71	24366.79	25008.58	24745.05
8-65-7	46389.98	45296.65	46216.56	45553.83	46749.3	46256.98
9-70-5	27257.63	25545.31	25532.28	25277.36	25450.91	25224.12
9-75-7	29256.86	29142.09	29272.45	29137.69	29205.54	29107.46
9-85-5	23355.22	23022.43	22858.58	22608.26	23240.07	22985.04
9-88-7	28497.24	28392.9	28615.11	28451.68	28676.73	28392.88
10-90-7	25744.83	25484.67	25438.81	25245.69	25437.12	25021.07
15-100-7	20676.81	18625.63	20581.55	20285.07	20507.03	20234.9
15-100-10	32742.03	31188.25	32454.69	31394.29	32586.89	31942.82
15-120-10	31712.23	30893.46	32617.02	32180.6	32171.38	31680.79
20-150-10	27011.78	26103.88	26916.86	26619.65	26723.04	26606.19
20-180-12	55894.64	55090.93	55474.01	55074.75	56836.62	56310.66
25-200-12	53660.57	52278.51	52938.72	52275.73	52322.95	51564.92
30-250-10	40608.62	39350.63	40342.18	39633.71	40846.64	39432.27
30-270-10	39793.64	37218.66	38271.97	37788.59	37477.92	36481.77
35-300-10	70530.91	67347.98	69935.89	69155.16	69935.89	69155.16
35-310-12	70366.98	67722.48	69916.42	69088.91	69916.42	69088.91
Average	29584.1	28808.95	**29335.93**	**29022.74**	29419.67	29029.51

The above results illustrate that both GVNS schemes using adaptive shaking perform better than the GVNS scheme using the classic intensified shaking. More specifically, both of the adaptive shaking methods are more effective than the classic one. This effectiveness may depend on the reduced randomness in the selection of shaking operators.The adaptive shaking with a complexity-based initial order is a pure deterministic method, while the second one confines randomness in the initial order of its operators. In the classical shaking method, each shaking operator has the same probability to be selected. It has been observed that in some problem instances one or more shaking operators cannot lead to efficient search, they keep being selected iteratively, though. Moreover, the $GVNS\_2$ produces better quality solutions than the $GVNS\_3$. Further, the $GVNS\_2$ is compared with its corresponding reduced scheme, $GVNS\_2_R$. Their numerical results are reported in Table 8.

Table 8: $GVNS\_2$ vs $GVNS\_2_R$

Instance	$GVNS\_2\_Avg$	$GVNS\_2\_Best$	$GVNS\_2_R\_Avg$	$GVNS\_2_R\_Best$
4-9-3	19965.42	19893.35	20001.45	19893.35
4-10-3	20388.18	20386.88	20371.25	20306.64
4-10-5	16654.57	16639.49	16670.78	16639.5
4-12-5	19330	19218.7	19408.15	19363.85
4-15-3	10204.27	10199.25	10202.1	10197.66
5-12-3	12977.22	12965.53	12965.53	12965.52
5-15-3	15978.63	15967.97	15975.89	15966.53
5-15-5	21973.37	21829.92	21973.37	21829.92
5-18-3	22393.16	22044.37	22912.03	22393.16
5-20-3	19097.41	18970.59	19060.16	19013.63
6-40-5	22033.75	21955.77	22054.8	21930.66
7-52-5	16492.06	16338.96	16431.74	16213.8
7-55-7	20289.07	20133.6	20263.26	20188.2
8-60-5	24917.71	24366.79	24917.71	24366.79
8-65-7	46216.56	45553.83	46550.04	45434.11
9-70-5	25532.28	25277.36	25260.02	25095.64
9-75-7	29272.45	29137.69	29229.07	29051.29
9-85-5	22858.58	22608.26	22977.73	22777.82
9-88-7	28615.11	28451.68	28594.76	28410.18
10-90-7	25438.81	25245.69	25599.85	25336.84
15-100-7	20581.55	20285.07	20670.1	20333.84
15-100-10	32454.69	31394.29	32454.69	31394.29
15-120-10	32617.02	32180.6	32303.52	31684.38
20-150-10	26916.86	26619.65	26928.55	26681.24
20-180-12	55474.01	55074.75	55474.01	55074.75
25-200-12	52938.72	52275.73	52965.83	52275.73
30-250-10	40342.18	39633.71	40497.02	39821.39
30-270-10	38271.97	37788.59	38190.2	37634.62
35-300-10	69935.89	69155.16	70789.39	69969.02
35-310-12	69916.42	69088.91	71584.55	68786.73
Average	**29335.93**	**29022.74**	29442.59	29034.37

The results indicate that the $GVNS\_2$ is a more suitable scheme for solving the homogeneous case of the problem than its reduced version. Thus, the $GVNS\_2$ is compared with the results obtained by the CPLEX solver, in order to further evaluate its efficiency. This comparison is summarized in Table 9. "OM" indicates the out-of-memory error occurred by solving large-sized instances.

Table 9: Compare the results achieved by $GVNS\_2$ and CPLEX (using light-duty vehicles)

Instance	CPLEX (a)	$GVNS\_2\_Avg$ **(b)**	$GVNS\_2\_Best$ **(c)**	**Gap a-b %**	**Gap a-c %**
4-9-3	19261.33	19965.42	19893.35	- 3.66	- 3.28
4-10-3	20022.66	20388.18	20306.64	- 1.83	- 1.82
4-10-5	16690.5	16654.57	16639.5	0.22	0.31
4-12-5	19551.98	19330	19218.7	1.14	1.7
4-15-3	10412.98	10204.27	10199.25	2	2.05
5-12-3	13146.48	12977.22	12965.53	1.29	1.38
5-15-3	15715.24	15978.63	15965.53	- 1.68	- 1.61
5-15-5	23045.4	21973.37	21829.92	4.65	5.27
5-18-3	22572.41	22393.16	22044.37	0.79	2.34
5-20-3	23873.07	19097.41	18970.59	20	20.54
6-40-5	N/A	22033.75	21955.77	-	-
7-52-5	N/A	16492.06	16338.96	-	-
7-55-7	N/A	20289.07	20133.6	-	-
8-60-5	N/A	24917.71	24366.79	-	-
8-65-7	N/A	46216.56	45553.83	-	-
9-70-5	N/A	25532.28	25277.36	-	-
9-75-7	N/A	29272.45	29137.69	-	-
9-85-5	N/A	22858.58	22608.26	-	-
9-88-7	N/A	28615.11	28594.76	-	-
10-90-7	OM	25438.81	25245.69	-	-
15-100-7	OM	20581.55	20285.07	-	-
15-100-10	OM	32454.69	31394.29	-	-
15-120-10	OM	32617.02	32180.6	-	-
20-150-10	OM	26916.86	26619.65	-	-
20-180-12	OM	55474.01	55074.75	-	-
25-200-12	OM	52938.72	52275.73	-	-
30-250-10	OM	40342.18	39633.71	-	-
30-270-10	OM	38271.97	37788.59	-	-
35-300-10	OM	69935.89	69155.16	-	-
35-310-12	OM	69916.42	69088.91	-	-

The $GVNS\_2$ produces almost 3% better solutions than CPLEX in average for the case of the small-sized instances. Focused on the best found solutions of the $GVNS\_2$, this gap is increased approximately to 3.4%. As it can be noticed, the CPLEX solver cannot provide feasible solutions for the medium-sized instances under the specified time limit. Moreover, an out-of-memory error occurred during the solution of the medium-sized instance "10-90-

7" and all large-sized instances. As the $GVNS\_2$ proved to be efficient in solving problem instances of the studied problem, it is also used to solve these instances under the usage of medium-duty trucks. The achieved results are compared with those produced by CPLEX solver and they are reported in Table 10.

Table 10: Compare the results achieved by $GVNS\_2$ and CPLEX (using medium-duty vehicles)

Instance	CPLEX (a)	$GVNS\_2\_Avg$ **(b)**	$GVNS\_2\_Best$ **(c)**	**Gap a-b %**	**Gap a-c %**
4-9-3	19161.21	19907.18	19867.3	- 3.89	- 3.68
4-10-3	19871.37	20025.72	19984.59	- 0.78	- 0.57
4-10-5	16641.27	16481.28	16478.75	0.96	0.98
4-12-5	20477.68	19135.32	19125.16	6.56	6.6
4-15-3	10296.25	10178.69	10176.11	1.14	1.17
5-12-3	13038.15	12858.72	12854.84	1.38	1.41
5-15-3	15633.85	15801.46	15787.22	- 1.07	- 0.98
5-15-5	23523.54	21838.86	20449.79	7.16	13.07
5-18-3	22345.25	21816.29	20858.26	2.37	6.65
5-20-3	20379.37	18857	18784.91	7.47	7.82
6-40-5	N/A	21401.83	21069.95	2.46	3.86
7-52-5	N/A	16370.36	16202.46	-	-
7-55-7	N/A	20684.1	20289.71	-	-
8-60-5	N/A	23977.64	23607.5	-	-
8-65-7	N/A	42803.82	41951.12	-	-
9-70-5	N/A	25551.22	24409.88	-	-
9-75-7	N/A	28340.91	28310.15	-	-
9-85-5	N/A	23452.48	23050.71	-	-
9-88-7	N/A	29048.78	28817.38	-	-
10-90-7	OM	25876.82	25483.62	-	-
15-100-7	OM	15211.71	13986.74	-	-
15-100-10	OM	32413.15	32058.96	-	-
15-120-10	OM	33093.59	32846.37	-	-
20-150-10	OM	27227.04	26836.73	-	-
20-180-12	OM	58273.09	57768.52	-	-
25-200-12	OM	53710.14	52649.32	-	-
30-250-10	OM	46402.38	45561.52	-	-
30-270-10	OM	39775.43	39323.89	-	-
35-300-10	OM	65889.8	64470.15	-	-
35-310-12	OM	73180.99	72344.19	-	-

The proposed GVNS algorithm performs approximately 2.5% better than CPLEX in solving the homogeneous case of the problem using medium-duty vehicles (approximately up to 4% focused on best found solutions of GVNS). Moreover, the commercial solver cannot provide feasible solutions even for some small-sized instances.

The impact of using different type of vehicles on fuel consumption, its cost and CO_2 emissions is illustrated in Figures 2, 3 and 4 respectively.

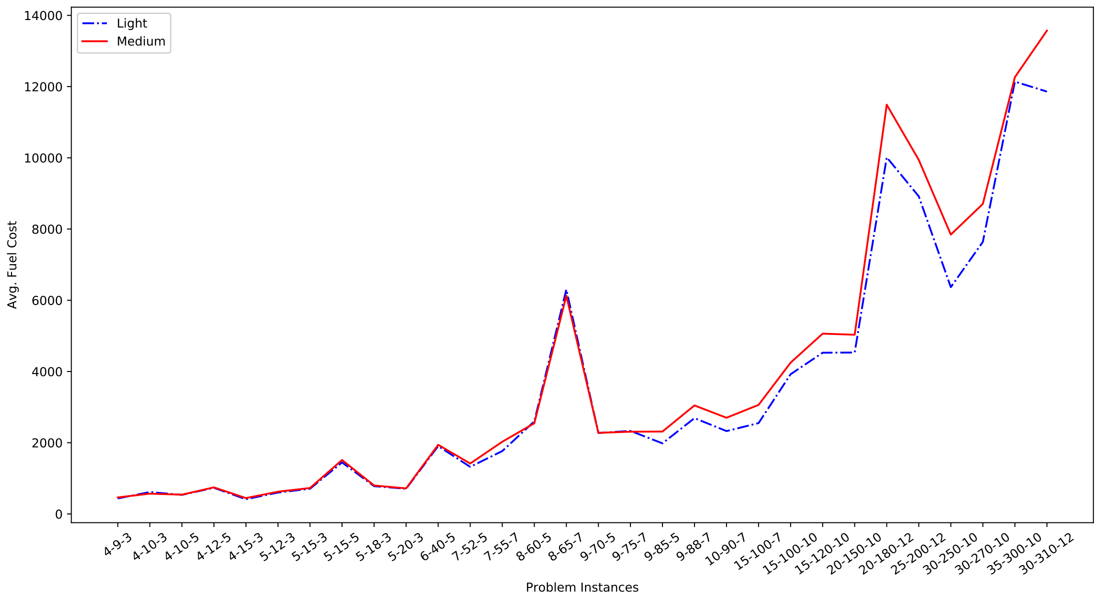


Figure 2: The average fuel consumption cost in cases of light- and medium-duty vehicles.

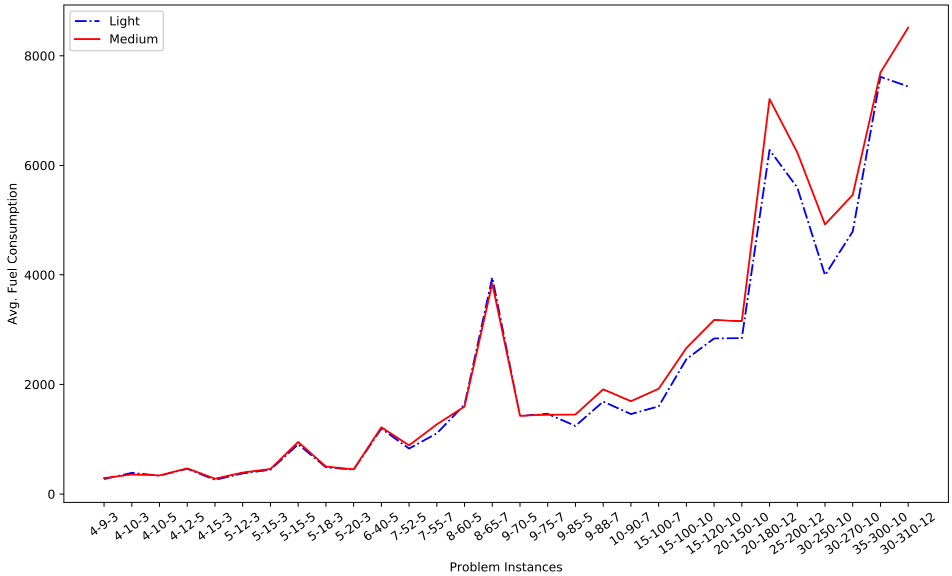


Figure 3: The average fuel consumption (L) in cases of light- and medium-duty vehicles.

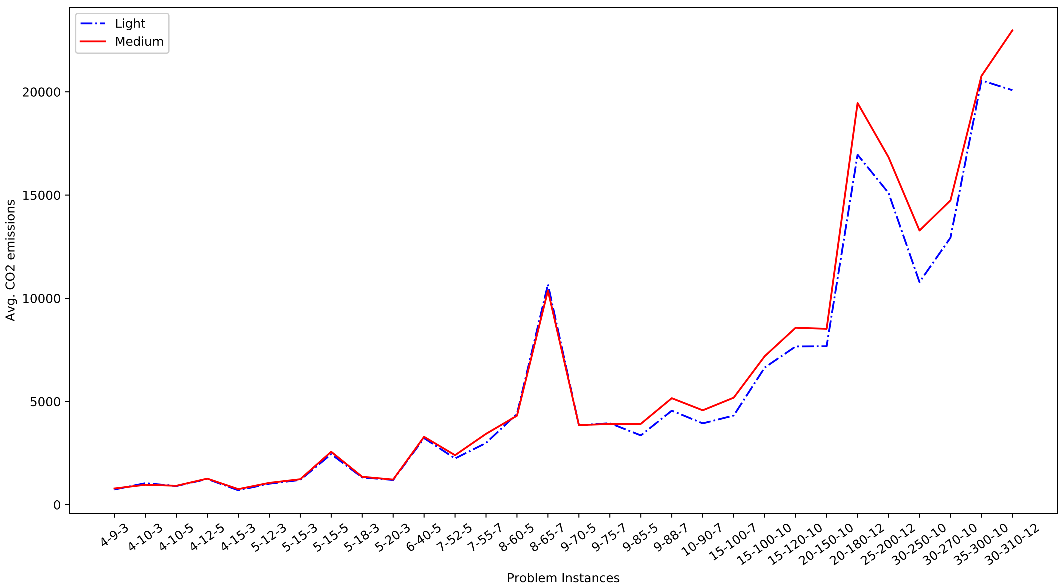


Figure 4: The average CO_2 emissions (kg) in cases of light- and medium-duty vehicles.

It is important to highlight that the use of medium-duty vehicles leads to significant decrease on fuel consumption and CO_2 emissions. More specifically, both the fuel consumption and the CO_2 levels are decreased by 2.6%. There is a twofold explanation for these reductions. First, by adopting a fleet with medium-duty vehicles, fewer and better-formed routes can be built than using light-duty trucks. Also, using medium-duty vehicles leads to a smaller fleet size than in the case of light-duty vehicles. Nonetheless, the fuel consumption cost is increased by 9.66% using medium-duty vehicles. This may be attributed by the total weight (curb and load weight) which is obviously increased in the case of medium-duty fleet of vehicles.

However, a mixed-fleet is commonly adopted in real-life applications. Thus, further examination is made in this direction. $GVNS\_4$, $GVNS\_5$ and $GVNS\_6$ are initially tested on the 30 random generated instances. Their results are provided in Table 11.

Table 11: Average and best values of $GVNS\_4$, $GVNS\_5$ and $GVNS\_6$

Instance	$GVNS_4\_Avg$	$GVNS_4\_Best$	$GVNS_5\_Avg$	$GVNS_5\_Best$	$GVNS_6\_Avg$	$GVNS_6\_Best$
4-9-3	30520.18	29647.29	30765.42	30693.35	30829.18	30799.61
4-10-3	35995.16	34513.94	34548.23	34513.94	36137.61	34513.95
4-10-5	28832	28832	28736.49	28714.56	28719.87	28714.62
4-12-5	37984.05	37211.55	37084.53	36260.56	39506.74	37204
4-15-3	17403.02	17397.8	17404.49	17397.68	17406.42	17397.69
5-12-3	27441.68	27423.38	27440.32	27422.85	27427.79	27422.85
5-15-3	30375.2	30368.01	30373.36	30366.56	30374.89	30368.03
5-15-5	69067.65	67497.91	68913.66	66977.66	69133.17	68662.37
5-18-3	46657.86	46390.79	46656.1	46390.98	46917.01	46406.27
5-20-3	33527.63	33475.16	33526.97	33470.58	33485.1	33402.98
6-40-5	61421.62	59524.58	62042.52	61639.27	61680.45	61591.17
7-52-5	40447.48	40391.73	40481.73	40448.62	40492.68	40428.09
7-55-7	45860.46	45718.6	45621.25	45421.78	45926.44	45722.02
8-60-5	94385.55	93461.52	93143.03	92208.35	93827.05	92065.25
8-65-7	253406.5	251405.2	253101.7	251508.6	252217.1	249853.3
9-70-5	73244.6	68938.83	72703.7	68663.82	72474.47	69907.24
9-75-7	58312.67	58174.97	58347.14	58260.16	58187.53	57971.01
9-85-5	47269.27	46890.23	47061.94	46992.61	47293.92	47010.44
9-88-7	62355.02	62181.18	62376.73	62196.86	62315.84	62242.41
10-90-7	45197.7	43846.87	43894.43	43761.46	43860.43	43517.92
15-100-7	50805.08	48844.84	50482.09	48101.82	48846.77	48371.12
15-100-10	71896.95	69419.84	71738.12	70931.28	71147.65	70017.36
15-120-10	78304.45	76205.23	76163.42	74377.12	78649.79	77165.66
20-150-10	87011.1	86023.12	86066.52	83902.64	86030.11	83007.77
20-180-12	187639.4	186855.5	187120.5	186712.4	186053.3	184981.3
25-200-12	165293	163618.9	164964.5	161578.4	164715.2	162377.6
30-250-10	76810.38	75669.81	76497.02	75821.38	77208.77	76601.8
30-270-10	88248.25	85529.95	86804.2	86043.47	86473.67	85925.92
35-300-10	196312	194688.7	196371.9	194698.6	194955.3	194174.2
35-310-12	133082	131259.6	133164.1	132237.7	133164.1	132237.7
Average	75836.93	74713.57	**75453.2**	**74590.5**	75515.28	74668.72

Similar to the homogeneous case of the problem, the GVNS scheme which uses the adaptive shaking mechanism with a complexity-based initial order is proved the most efficient method. Furthermore, it is interesting to examine the reduced case of $GVNS\_5$. Table 12 contains the average and the best found solutions of $GVNS\_5_R$ and their gap (%) from the corresponding $GVNS\_5$ solutions.

Table 12: The results achieved by $GVNS\_5_R$ and their gap from the results of $GVNS\_5$

Instance	$GVNS\_5_R\_Avg$	$GVNS\_5_R\_Best$	**Gap Avg. Solutions %**	**Gap Best Solutions %**
4-9-3	30786.67	30693.35	-0.07	0
4-10-3	34487.43	34381.39	0.18	0.38
4-10-5	28776.89	28731.59	-0.14	-0.06
4-12-5	37093.05	36277.87	-0.02	-0.05
4-15-3	17405.14	17399.35	0	-0.01
5-12-3	27432.32	27422.85	0.03	0
5-15-3	30379.59	30368.05	-0.02	0
5-15-5	69218.02	67989.64	-0.44	-1.51
5-18-3	46775.61	46390.61	-0.26	0
5-20-3	33520.76	33481.52	0.02	-0.03
6-40-5	61423.02	60347.19	1	2.1
7-52-5	40679.31	40458.96	-0.49	-0.03
7-55-7	45511.59	45452.61	0.24	-0.07
8-60-5	93115.1	92390.42	0.03	-0.2
8-65-7	251383.6	249532.1	0.68	0.79
9-70-5	72597.73	69526.91	0.15	-1.26
9-75-7	58223.48	58030.34	0.21	0.39
9-85-5	47107.16	46879.55	-0.1	0.24
9-88-7	62397.38	62219.98	-0.03	-0.04
10-90-7	43938.91	43811.77	-0.1	-0.11
15-100-7	45505.46	41004.02	9.86	14.76
15-100-10	72098.84	71548.5	-0.5	-0.87
15-120-10	76792.02	76115.48	-0.83	-2.34
20-150-10	86059.3	84094.49	0.01	-0.23
20-180-12	186510	185948.6	0.33	0.41
25-200-12	163087.7	159319.8	1.14	1.4
30-250-10	76439.84	75821.38	0.07	0
30-270-10	86554.11	85849.48	0.29	0.23
35-300-10	195684.7	194490.1	0.35	0.11
35-310-12	133164.1	132237.7	0	0

Despite the results clearly indicate that both $GVNS\_5$ and $GVNS\_5_R$ perform almost equivalently, it seems that the GVNS with the reduced adaptive shaking scheme can produce slightly better solutions than the initial scheme, especially on large problem instances. This may be occurred by the significant reduction of shaking iterations which enables the improvement phase to be executed more times. Furthermore, during the experiments with $GVNS\_5$ and $GVNS\_5_R$, it is noticed that the solutions with an increase in vehicle mixing, are found to be the best. In this direction, an alternative of the $GVNS\_5$ and $GVNS\_5_R$ schemes ($GVNS\_5^*$ and $GVNS\_5^*_R$ respectively), which use the $Shake_2$ instead of $Shake_1$, are tested. Due to the fact that a local search operator is more complex than a shaking operator, the shaking operator S_5 is selected to be used as the expedient on increasing the fleet diversity. However, in order to control this diversity, operator N_8 is also used. The numerical results of $GVNS\_5^*$ and $GVNS\_5^*_R$ are given in Table 13.

Table 13: The average and best found results of $GVNS\_5^*$ and $GVNS\_5^*_R$

Instance	$GVNS\_5^*\_Avg$ **(b)**	$GVNS\_5^*\_Best$ **(c)**	$GVNS\_5^*_R\_Avg$	$GVNS\_5^*_R\_Best$
4-9-3	30556.21	29647.29	30828.79	30723.82
4-10-3	34302.87	33435.3	34759.74	34383.76
4-10-5	29062.39	28716.11	29091.07	28713.12
4-12-5	33141.53	31853.81	33695.88	31857.18
4-15-3	17402.1	17397.66	17405.18	17402.33
5-12-3	27633.45	27423.38	27390.11	27136.26
5-15-3	29351.3	28505.29	29388.97	28782.4
5-15-5	67118.98	65760.69	68626.54	65987.51
5-18-3	44665.75	43821.99	45368.95	44717.53
5-20-3	33515.91	33409.03	33578.46	33375.17
6-40-5	62363.02	60861.09	61515.82	59792.91
7-52-5	40515.12	40408.63	40571.34	40364.73
7-55-7	45617.8	45369.76	45614.4	45435.75
8-60-5	93676.57	92339.34	93438.79	92419.65
8-65-7	252522.6	251273.5	253098.5	251052.1
9-70-5	74397.69	71337.26	72309.77	68820.45
9-75-7	58347.14	58260.16	58187.02	57950.38
9-85-5	47023.98	46917.3	47088.62	46879.55
9-88-7	62373.95	62196.86	62389.43	62233.86
10-90-7	44038.39	43900.62	43981.29	43738.73
15-100-7	49374.27	48124.49	49237.94	48022.45
15-100-10	71331.44	70797.74	71560.47	71225.41
15-120-10	76893.95	74216.33	77064.65	75899.13
20-150-10	85949.9	83769.98	86057.27	83902.64
20-180-12	187129.2	186536.6	187238.5	186772.4
25-200-12	162581	159529	165767.3	164258.2
30-250-10	76497.02	75821.38	76439.84	75821.38
30-270-10	86813.8	85990.55	86723.59	86022.63
35-300-10	195334.7	194490.1	196106.5	194698.6
35-310-12	132730.1	131412.7	132730.1	131412.7
Average	75075.4	74117.46	75241.83	74326.76

From the reported results in Table 13, it is observed that the strategy of increasing fleet diversity leads to further improvements, as the fleet mixing can potentially lead to better formation of routes and lower vehicles usage costs. Also, following this approach, the $GVNS\_5^*$ performs slightly better than its reduced variant both in terms of average and best found solutions.

To further evaluate, the performance of $GVNS\_5^*$, a comparison with CPLEX is attempted and results are provided in Table 14.

Table 14: Compare the results achieved by $GVNS\_5^*$ and CPLEX

Instance	CPLEX (a)	$GVNS\_5^*\_Avg$ **(b)**	$GVNS\_5^*\_Best$ **(c)**	**Gap a-b %**	**Gap a-c %**
4-9-3	30303	30556.21	29647.29	- 0.84	2.16
4-10-3	33457.4	34302.84	33435.3	- 2.53	0.07
4-10-5	32475.16	29062.39	28716.11	10.51	11.58
4-12-5	38363.81	33141.53	31853.81	13.61	16.97
4-15-3	18356.11	17402.1	17397.66	5.2	5.22
5-12-3	28593.7	27633.45	27423.38	3.36	4.09
5-15-3	N/A	14715.93	14708.42	-	-
5-15-5	N/A	18128.85	18120	-	-
5-18-3	N/A	20342.34	19817.5	-	-
5-20-3	N/A	17786.57	17760.79	-	-
6-40-5	N/A	18515.07	18316.83	-	-
7-52-5	N/A	14316.28	14247.63	-	-
7-55-7	N/A	17385.93	17327.99	-	-
8-60-5	N/A	20181.07	19868.29	-	-
8-65-7	N/A	31585.7	31448.2	-	-
9-70-5	N/A	21194.36	21056.55	-	-
9-75-7	N/A	23521.41	23486.1	-	-
9-85-5	N/A	19425.33	19172.77	-	-
9-88-7	N/A	24115.2	24003.46	-	-
10-90-7	OM	22148.5	21964.04	-	-
15-100-7	OM	10337.47	10140.74	-	-
15-100-10	OM	25651.77	25471.85	-	-
15-120-10	OM	25090.81	24708.4	-	-
20-150-10	OM	19229.42	19055.34	-	-
20-180-12	OM	39357.65	38958.88	-	-
25-200-12	OM	36297.77	35844.38	-	-
30-250-10	OM	31506.05	31080.7	-	-
30-270-10	OM	23175.82	22903.12	-	-
35-300-10	OM	48197.6	47398.64	-	-
35-310-12	OM	47498.46	47010.09	-	-

As shown in Table 14, CPLEX can produce feasible solutions only for the six out of ten small-sized instances. The $GVNS\_5^*$ performs approximately 5.21% better than CPLEX, and their difference is increased up to around 7.2% in the case of best found solutions by $GVNS\_5^*$. Considering the high complexity of the studied problem, the achieved quality difference and the significant difference on execution time limits, it can be highlighted that the proposed GVNS scheme is quite efficient for solving the FSMPLIRP. Despite CPLEX is a state-of-the-art optimization solver, setting a strict time limit for the solution of NP-hard problems leads to the production of solutions with high optimality gap (gap between the best integer and the relaxed LP solution). Thus, the solutions obtained by our proposed solution approach are better even for small-sized instances.

4.4. Medium-duty vehicles vs mixed-fleet

It has been shown that the use of medium-duty only vehicles performs much better than using only light-duty vehicles mainly in terms of fuel consumption and CO_2 emissions. Therefore, it is interesting to examine how a homogeneous fleet of medium-duty vehicles and a mixed-fleet affect the fuel consumption (L), the corresponding cost and the CO_2 emissions (kg). Figures 5, 6 and 7 illustrate the discussed impacts for various problem instances.

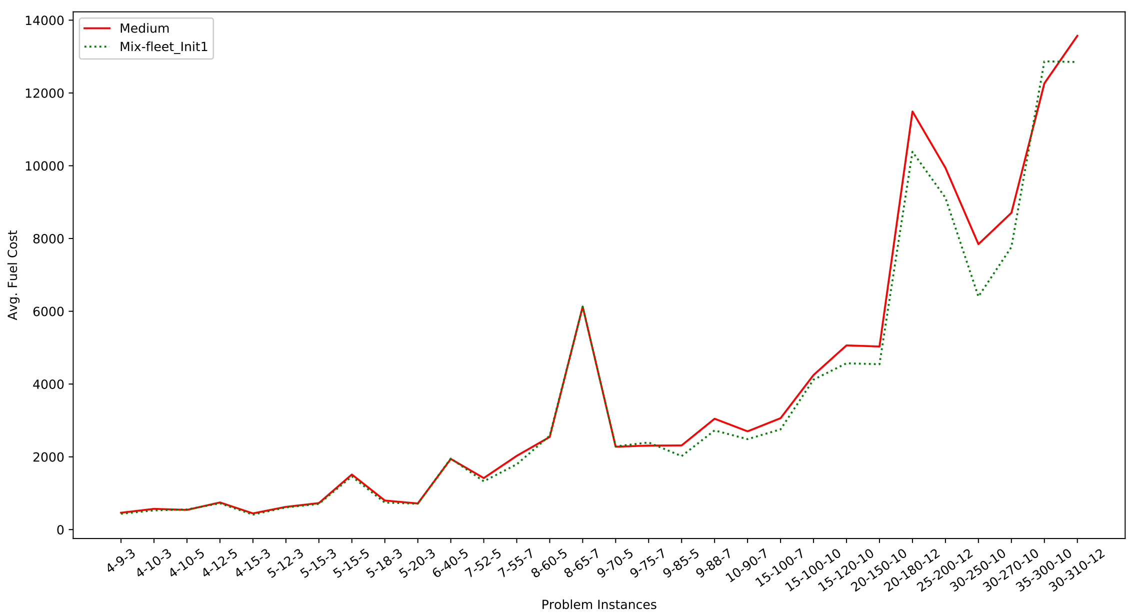


Figure 5: The average fuel consumption cost in cases of medium-duty vehicles and mixed-fleet.

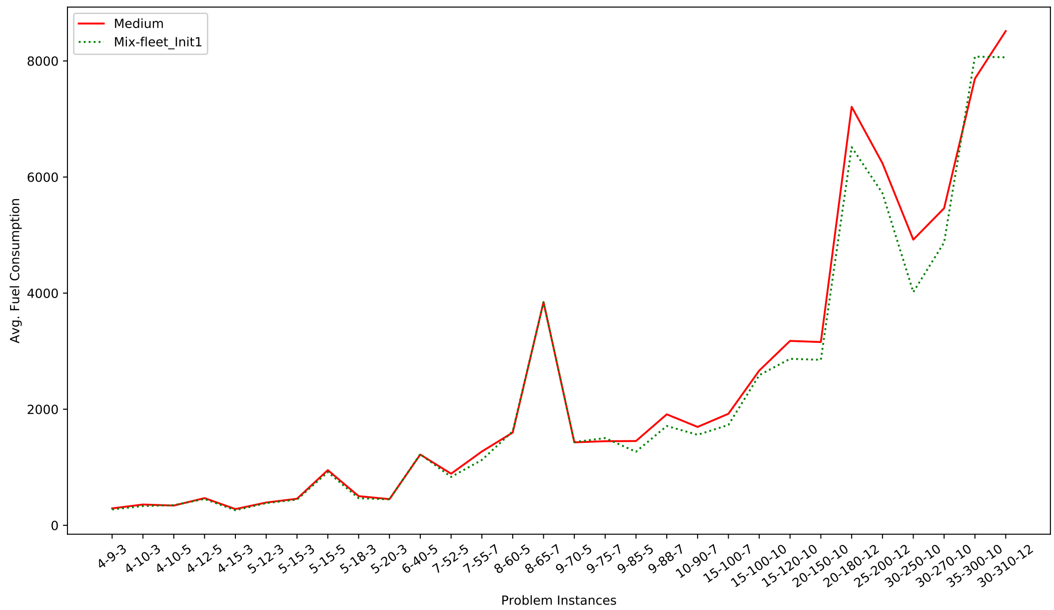


Figure 6: The average fuel consumption (L) in cases of medium-duty vehicles and mixed-fleet.

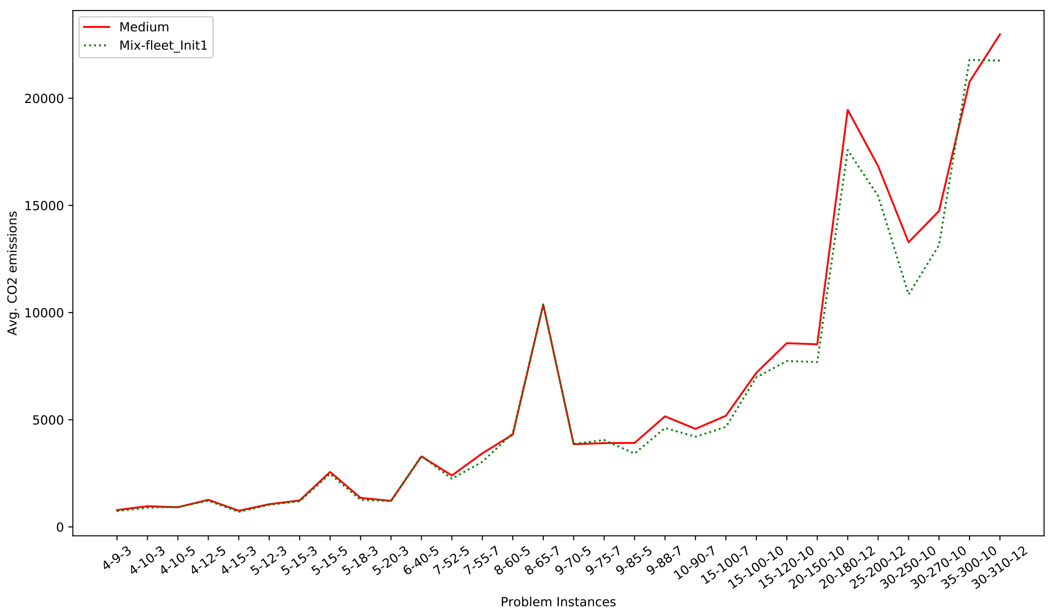


Figure 7: The average CO_2 emissions (kg) in cases of medium-duty vehicles and mixed-fleet.

The selection of a mixed-fleet significantly decreases the fuel consumption, the CO_2 emissions and their corresponding cost, especially for the case of large problem cases. However, an other critical decision parameter is the vehicle usage cost. Figure 8 illustrates the different vehicle usage cost levels for each fleet case.

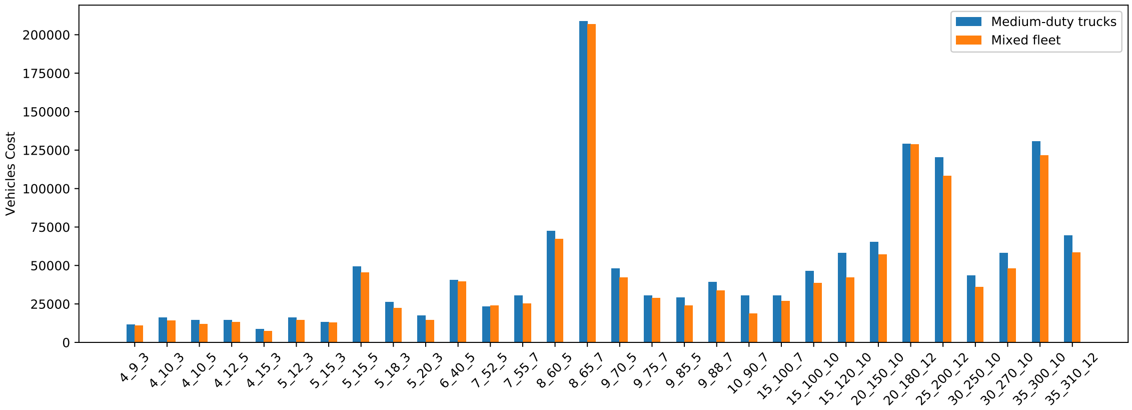


Figure 8: The vehicles usage costs in cases of medium-duty vehicles and mixed-fleet.

It is clear that the mixed-fleet is more cost effective than the case of using a homogeneous fleet of medium-duty vehicles (approximately 10%). Therefore, the use of a mixed-fleet is a sustainable strategic decision.

The impact of initialization. The use of different initialization rules has a potential effect on the solution of a GVNS heuristic (Hansen & Mladenović,2014). Therefore, it is examined whether an alternative customers' allocation rule has a considerable effect on the final solution of the $GVNS\_5^*$ (which has been proved the best scheme for solving the FSMPLIRP) or not. More specifically, the alternative allocation is also a nearest allocation method, which is applied by considering all the opened depots. The new GVNS scheme is mentioned as $GVNS\_5^*_{Init2}$. Table 15 provides the results obtained by $GVNS\_5^*_{Init2}$ and the comparison of them with the solutions produced by CPLEX.

Table 15: Compare the results achieved by $GVNS\_5^*$ using different initialization methods

Instance	$GVNS\_5^*_{Init2}\_Avg$	$GVNS\_5^*_{Init2}\_Best$	**Gap a %**	**Gap b %**
4-9-3	29992.57	29257.67	1.84	1.31
4-10-3	34727.45	34366.66	-1.24	-2.79
4-10-5	28928.39	29822.83	0.46	-0.72
4-12-5	33404.32	32841.5	-0.79	-3.1
4-15-3	17406.08	17400.79	-0.02	-0.02
5-12-3	27428.02	27423.36	0.74	0
5-15-3	29628.76	28672.01	-0.95	-0.58
5-15-5	70556.1	69437.41	-5.12	-5.59
5-18-3	47051.2	46479.4	-5.34	-6.06
5-20-3	33640.74	33528.41	-0.37	-0.36
6-40-5	63186.84	63129.26	-1.32	-3.73
7-52-5	39992.37	39828.16	1.29	1.44
7-55-7	45121.02	45020.08	1.09	0.77
8-60-5	94286.66	92886.35	-0.65	-0.59
8-65-7	249722.7	248333.5	1.11	1.17
9-70-5	72037.95	71681.4	3.17	-0.48
9-75-7	51536.19	51314.25	11.67	11.92
9-85-5	46006.43	45953.77	2.16	2.05
9-88-7	64279.82	63211.97	-3.06	-1.63
10-90-7	44038.39	43900.62	0	0
15-100-7	36894.5	32459.81	25.28	32.55
15-100-10	68180.36	67422.41	4.42	4.77
15-120-10	82139.7	81899.34	-6.82	-10.35
20-150-10	85949.9	83769.98	0	0
20-180-12	187034.1	186467.2	-0.05	0.04
25-200-12	148209	147419.9	8.84	7.59
30-250-10	78958.62	78463.9	-3.22	-3.49
30-270-10	89426.97	88609.33	-3.01	-3.05
35-300-10	182873.9	181887.7	6.38	6.48
35-310-12	115865	115465	12.71	12.14

The obtained results accentuate the impact of using different initialization rules. Focused on the large-sized instances the quality gap between $GVNS\_5^*$ and $GVNS\_5^*_{Init2}$ is approximately 5%. It has been observed that, by applying this alternative initialization rule, better routes can be built. The potential efficient geographic segmentation of customers can be a reasonable justification of the reported improvements. It is also interesting to focus on the potential effect of different initialization methods on fuel- and emissions-based details. Figures 9, 11 and 12 illustrate the observed differences.

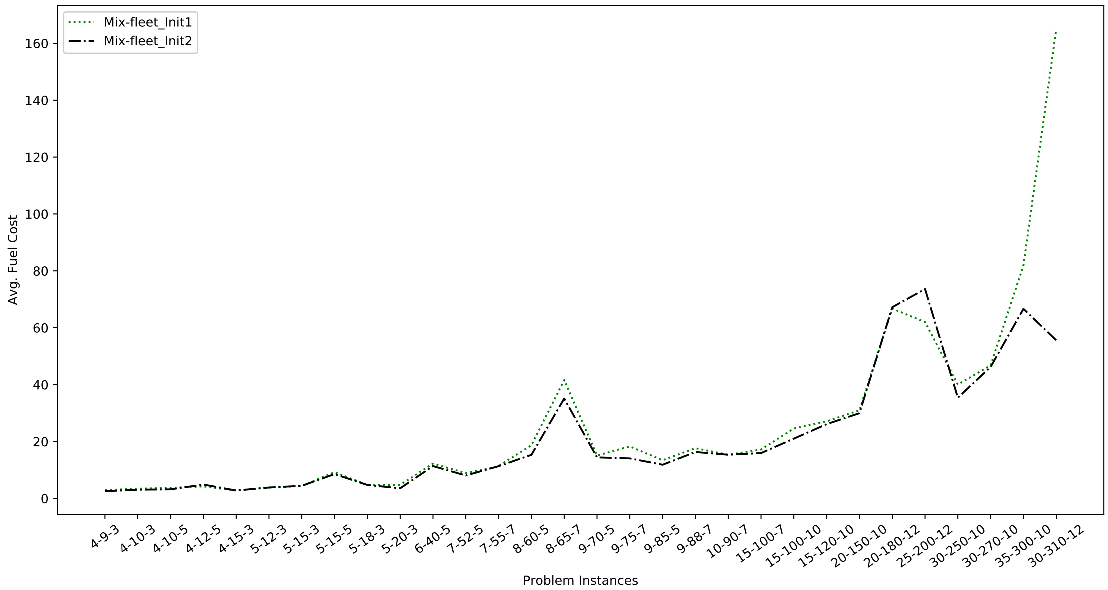


Figure 9: The average fuel consumption cost using different initialization rules.

Figure 9 cannot provide a clear view on fuel cost changes for the case of the ten small-sized instances. Thus, a more focused view on these instances is given in Figure 10.

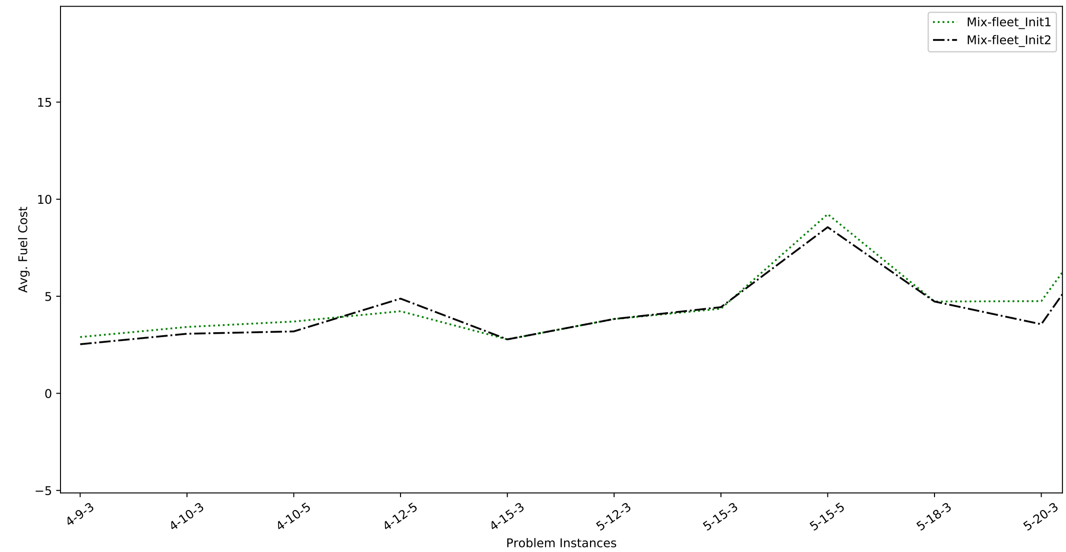


Figure 10: The average fuel consumption cost of ten small-sized instances using different initialization rules.

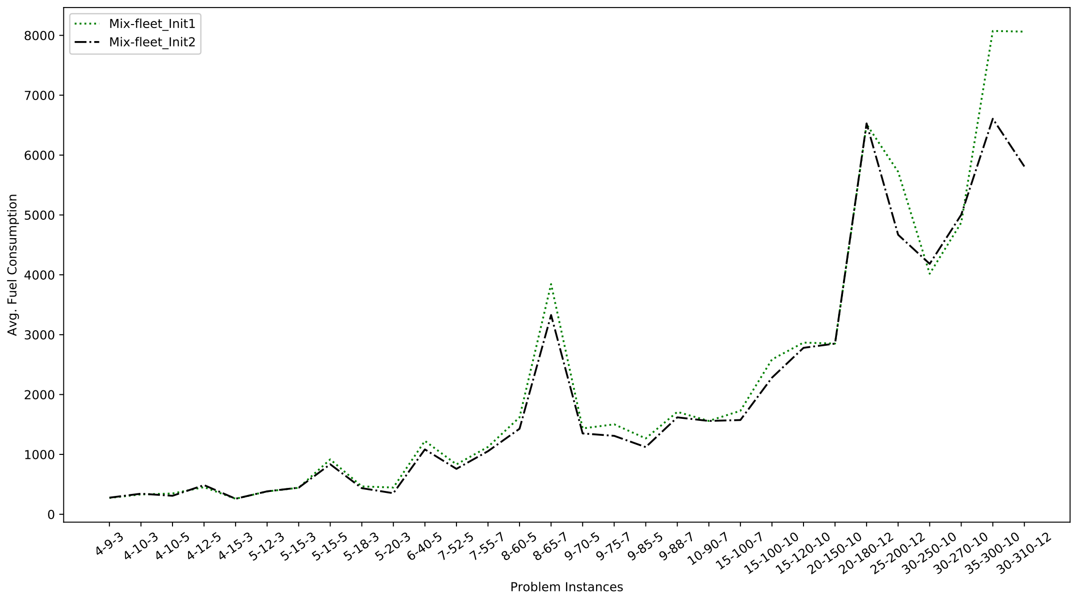


Figure 11: The average fuel consumption (L) using different initialization rules.

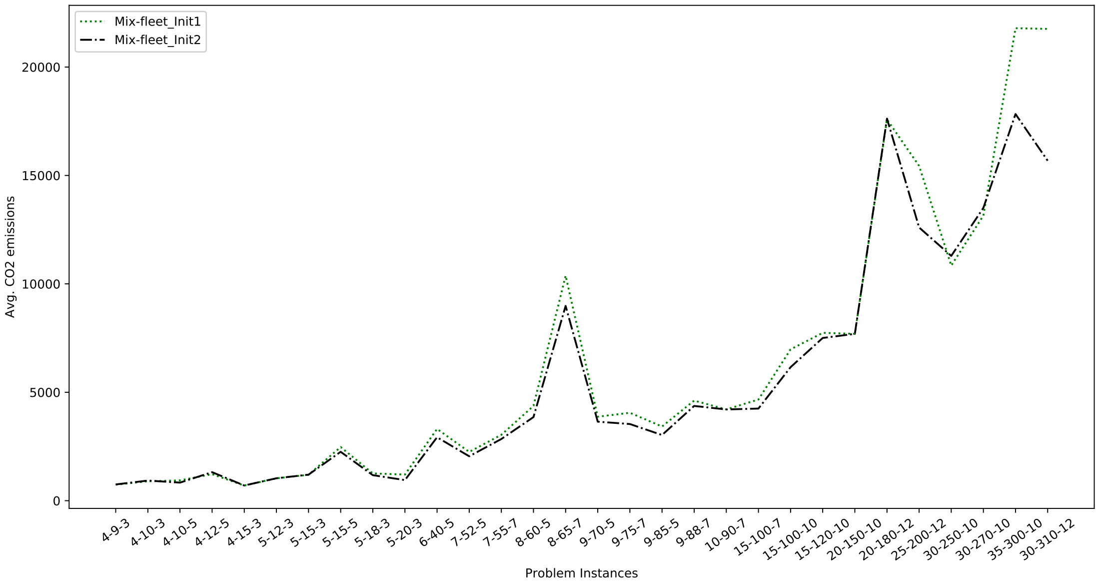


Figure 12: The average CO_2 emissions (kg) using different initialization rules.

It is noted that the $GVNS\_5^*_{Init2}$ mainly leads to more environmentally efficient solutions.

4.5. Opened depots and fleet composition

This section summarizes the number of opened depots, the capacity levels and the number (and type) of vehicles as they reported in the best found solutions for each problem instance. Thus, Table 16 provide the number of opened depots for each case of the studied problem and each solution method (CPLEX & GVNS).

Table 16: Number of opened depots per instance.

Instance	CPLEX_Light	$GVNS\_2$ (Light)	CPLEX_Medium	$GVNS\_2$ (Medium)	CPLEX_MixedFleet	$GVNS\_5^{*}_{Init2}$
4-9-3	2	2	2	2	2	2
4-10-3	2	2	2	2	2	2
4-10-5	2	2	2	2	2	2
4-12-5	3	2	2	2	2	2
4-15-3	1	1	1	1	2	1
5-12-3	1	1	1	1	1	1
5-15-3	1	1	1	1	-	1
5-15-5	2	2	2	2	-	2
5-18-3	2	2	2	2	-	2
5-20-3	3	2	-	2	-	2
6-40-5	-	2	-	2	-	2
7-52-5	-	2	-	2	-	2
7-55-7	-	2	-	2	-	2
8-60-5	-	2	-	2	-	2
8-65-7	-	2	-	2	-	2
9-70-5	-	2	-	2	-	2
9-75-7	-	2	-	2	-	2
9-85-5	-	2	-	2	-	2
9-88-7	-	2	-	2	-	2
10-90-7	-	2	-	2	-	2
15-100-7	-	2	-	2	-	2
15-100-10	-	2	-	2	-	2
15-120-10	-	2	-	2	-	2
20-150-10	-	2	-	2	-	2
20-180-12	-	1	-	1	-	1
25-200-12	-	2	-	2	-	2
30-250-10	-	2	-	2	-	2
30-270-10	-	2	-	2	-	2
35-300-10	-	2	-	2	-	2
35-310-12	-	2	-	2	-	2

Solution by the proposed GVNS-based heuristic algorithms lead to opening the minimum required number of depots. As shown in Table 16, the proposed GVNS algorithms managed to open equal or less depots than the CPLEX solver for the case of small-sized instances. In all problem cases the same depots are selected to be opened. The reason behind this fact, is that in all these cases the structure of locations are kept unmodified. A more detailed information about the opened depots and their planned capacity levels is given in Table 17.

Table 17: The opened depots and their capacity levels.

Instance	Decisions	Values		Instance	Decisions	Values	
4-9-3	depots	depot_2	depot_3	9-70-5	depots	depot_4	depot_5
	cap. level	level_2	level_3		cap. level	level_3	level_3
4-10-3	depots	depot_3	depot_4	9-75-7	depots	depot_3	depot_9
	cap. level	level_2	level_1		cap. level	level_2	level_3
4-10-5	depots	depot_1	depot_3	9-85-5	depots	depot_3	depot_9
	cap. level	level_2	level_2		cap. level	level_4	level_4
4-12-5	depots	depot_1	depot_4	9-88-7	depots	depot_1	depot_2
	cap. level	level_1	level_2		cap. level	level_3	level_2
4-15-3	depots	depot_4		10-90-7	depots	depot_2	depot_4
	cap. level	level_2			cap. level	level_1	level_3
5-12-3	depots	depot_1		15-100-7	depots	depot_8	depot_9
	cap. level	level_3			cap. level	level_2	level_4
5-15-3	depots	depot_5		15-100-10	depots	depot_7	depot_14
	cap. level	level_2			cap. level	level_2	level_4
5-15-5	depots	depot_4	depot_5	15-120-10	depots	depot_9	depot_15
	cap. level	level_1	level_1		cap. level	level_3	level_4
5-18-3	depots	depot_1	depot_3	20-150-10	depots	depot_1	depot_11
	cap. level	level_3	level_2		cap. level	level_3	level_5
5-20-3	depots	depot_1	depot_4	20-180-12	depots	depot_14	
	cap. level	level_1	level_2		cap. level	level_2	
6-40-5	depots	depot_3	depot_6	25-200-12	depots	depot_11	depot_13
	cap. level	level_2	level_1		cap. level	level_5	level_5
7-52-5	depots	depot_4	depot_6	30-250-10	depots	depot_4	depot_20
	cap. level	level_3	level_2		cap. level	level_3	level_3
7-55-7	depots	depot_3	depot_7	30-270-10	depots	depot_13	depot_27
	cap. level	level_2	level_4		cap. level	level_2	level_2
8-60-5	depots	depot_3	depot_6	35-300-10	depots	depot_16	depot_24
	cap. level	level_2	level_2		cap. level	level_1	level_4
8-65-7	depots	depot_2	depot_5	35-310-12	depots	depot_26	depot_29
	cap. level	level_3	level_3		cap. level	level_4	level_3

Table 18 provides the fleet composition for the mixed-fleet problem case as it has been obtained by CPLEX solver for some of the small-sized instances and the $GVNS\_5^{*}_{Init2}$ for all problem instances. The fleet composition decided by $GVNS\_5^{*}_{Init2}$, corresponds to the best found solution for each problem instance. The letter "L" means light-duty vehicle and the letter "M" is used for medium-duty vehicles.

Table 18: The fleet composition by each method.

Instance	CPLEX	$GVNS\_5^{*}_{Init2}$	Instance	CPLEX	$GVNS\_5^{*}_{Init2}$
4-9-3	3 L & 1 M	2 L & 2 M	9-70-5	-	8 L
4-10-3	3 L & 3 M	4 L & 4 M	9-75-7	-	3 L
4-10-5	2 L & 1 M	2 L & 1 M	9-85-5	-	4 L
4-12-5	3 L & 1 M	2 L & 2 M	9-88-7	-	5 L
4-15-3	2 L & 1 M	2 L	10-90-7	-	1 L & 1 M
5-12-3	4 L & 1 M	5 L	15-100-7	-	2 L & 1 M
5-15-3	-	4 L & 3 M	15-100-10	-	2 L & 1 M
5-15-5	-	10 L	15-120-10	-	4 L & 1 M
5-18-3	-	8 L	20-150-10	-	5 L & 1 M
5-20-3	-	4 L & 1 M	20-180-12	-	9 L & 1 M
6-40-5	-	7 L	25-200-12	-	8 L & 1 M
7-52-5	-	4 L	30-250-10	-	4 L & 1 M
7-55-7	-	3 L	30-270-10	-	3 L & 1 M
8-60-5	-	13 L & 2 M	35-300-10	-	10 L
8-65-7	-	28 L	35-310-12	-	4 L & 1 M

Despite the efficiency of the proposed solution methods, a few limitations of this work should be mentioned. First, an alternative initial order of shaking operators in the adaptive shaking mechanisms may lead to further improvements. Moreover, focused on the strength of the shaking, five different values were examined. Further improvements can potentially achieved by investigating other values. Finally it is not possible to formally assess the quality of the obtained solution with respect to the truly optimal.

5. Conclusions

Sustainability is a crucial factor of a company's growth. In this regard, this work studies a new complex supply chain network optimization problem, which integrates both economic and environmental decisions. As commercial solvers cannot solve realistic cases of such complex problems, GVNS-based heuristic algorithms were developed for solving medium- and large-sized instances. The shaking mechanism in a VNS-based heuristic has a significant role in its performance. Thus, new adaptive shaking techniques are proposed, as a crucial intelligent learning component of the proposed solution method. This intelligent mechanism uses past experience in order to improve the performance of the algorithm. In these shaking methods, the shaking operators are ordered following two different rules. According to the first one, the operators set in an order, based on their complexity, while in the second one their ordering is performed randomly. During the execution of the algorithms, the shaking operators are re-ordered in accordance with the number of improvements achieved by using each of them in the previous iteration. The GVNS schemes using the proposed adaptive shaking mechanisms are proved more efficient on the solution of such complex supply chain network optimization problems than the GVNS using the classic intensified shaking. Furthermore, the impact of using homogeneous fleet (either light- or medium-duty vehicles) and mixed-fleet is examined not only from an economic perspective, but also from an environmentally point of view. A computational analysis illustrates that by using a mixed-fleet both economical and environmental benefits can be achieved. The impact of using an alternative initialization rule is also investigated and the obtained solutions, especially on ten large-sized instances, were further improved by 5%. The results from the extended numerical analysis illustrate the integration of the proposed models and solution techniques in an intelligent tool which can assist decision makers to derive fast and reliable decisions for the optimal design and operation of complex supply chains.

The efficiency of the proposed solution methods which use intelligent learning shaking mechanisms, highlights promising future work directions. One could focus on the implementation of a parallel version of the proposed algorithms in order to accelerate the intelligent re-ordering of the shaking operators. Also, the development of adaptive improvement mechanisms and alternative local search and shaking operators constitutes a promising research direction for the solution of such complex optimization problems of practical interest. Additionally, a systematic combination of the proposed solution methods with exact MIP techniques is expected to lead to a powerful matheuristic approach with improved solutions. The extension of the MIP model to account for further practical features, such as intermediate stops for refueling purposes and delivery lead times, is also another direction for future work.

References

Aldis, J. (2017). Strategic facility location, capacity acquisition, and technology choice decisions under demand uncertainty: Robust vs. non-robust optimization approaches. *European Journal of Operational Research*, *260*, 1095–1104.

Asadi, E., Habibi, F., Nickel, S., & Sahebi, H. (2018). A bi-objective stochastic location-inventory-routing model for microalgae-based biofuel supply chain. *Applied Energy*, *228*, 2235–2261.

Bektaş, T., J.F., E., Psaraftis, H., & Puchinger, J. (2019). The role of operational research in green freight transportation. *European Journal of Operational Research*, *274*, 807–823.

Bezerra, S. N., de Souza, S., & Souza, M. J. F. (2018). A GVNS algorithm for solving the multi-depot vehicle routing problem. *Electronic Notes in Discrete Mathematics*, *66*, 167–174.

Brooke, A., Kendrik, D., Meeraus, A., Raman, R., & Rosenthal, R. (1998). *GAMS-A Users Guide*.

Chen, D., Sun, G., & 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.

Cheng, C., Yang, P., Qi, M., & Rousseau, L. (2017). Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E*, *97*, 97–112.

De Armas, J., Melián-Batista, B., Moreno-Pérez, J., & Brito, J. (2015). GVNS for a real-world rich vehicle routing problem with time windows. *Engineering Applications of Artificial Intelligence*, *42*, 45–56.

Dukkanci, O., Kara, B. Y., & Bektaş, T. (2019). The green location-routing problem. *Computers & Operations Research*, *105*, 187–202.

Flood, M. (1956). The Traveling-Salesman Problem. *Operations Research*, *4*, 61–75.

Foo, P., Lee, V., Tan, G., & Ooi, K. (2018). A gateway to realising sustainability performance via green supply chain management practices: A PLS-ANN approach. *Expert Systems with Applications*, *107*, 1–14.

Fuqing, Z., Shuo, Q., Yi, Z., Weimin, M., Chuck, Z., & Houbin, S. (2019). A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem. *Expert Systems with Applications*, *126*, 321–339.

Ghorbani, A., & 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.

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

Habibi, F., Asadi, E., & Sadjadi, S. (2018). A location-inventory-routing optimization model for cost effective design of microalgae biofuel distribution system: A case study in Iran. *Energy Strategy Reviews*, *22*, 82–03.

Hansen, P., & Mladenović, N. (2014). Variable Neighborhood Search. In E. Burke, & G. Kendall (Eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 313–337). New York: Springer Science+Business Media.

Hansen, P., Mladenović, N., Brimberg, J., & Pérez, J. (2010). Variable Neighborhood Search. In M. Gendreau, & J. Potvin (Eds.), *Handbook of Metaheuristics* (pp. 61–86). Boston: International Series in Operations Research & Management Science.

Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization*, *5*, 423–454.

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

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

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2019a). A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing. *Computers & Chemical Engineering*, *126*, 263–279.

Karakostas, P., Sifaleras, A., & Georgiadis, M. C. (2019b). Basic VNS algorithms for solving the pollution location inventory routing problem. In A. Sifaleras, S. Salhi, & J. Brimberg (Eds.), *Variable Neighborhood Search (ICVNS 2018)* (pp. 64–76). Springer, Cham, LNCS volume 11328.

Koç, C., Bektaş, T., Jabali, O., & Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B*, *70*, 239–254.

Koç, C., Bektaş, T., Jabali, O., & Laporte, G. (2016). The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B*, *84*, 81–102.

Kong, L., Li, H., Luo, H., Ding, L., & Zhang, X. (2018). Sustainable performance of just-in-time (JIT) management in time-dependent batch delivery scheduling of precast construction. *Journal of Cleaner Production*, *193*, 684–701.

Lai, X., & Hao, J. (2016). Iterated variable neighborhood search for the capacitated clustering problem. *Engineering Applications of Artificial Intelligence*, *56*, 102–120.

Li, J., Wang, D., & Zhang, J. (2018). Heterogeneous fixed fleet vehicle routing problem based on fuel and carbon emissions. *Journal of Cleaner Production*, *201*, 896–908.

Lin, C., Choy, K., Ho, G., Chung, S., & Lam, H. (2014a). Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications*, *41*, 1118–1138.

Lin, C., Choy, K., Ho, G., & Ng, T. (2014b). A Genetic Algorithm-based optimization model for supporting green transportation operations. *Expert Systems with Applications*, *41*, 3284–3296.

Liu, B., Chen, H., Li, Y., & 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*, . doi:10.1155/2015/586581.

Liu, S., & 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.

Martins, C., & Pato, M. (2019). Supply chain sustainability: A tertiary literature review. *Journal of Cleaner Production*, *225*, 995–1016.

Micheli, J., & Mantella, F. (2018). Modelling an environmentally-extended inventory routing problem with demand uncertainty and a heterogeneous fleet under carbon control policies. *International Journal of Production Economics*, *204*, 316–327.

Mikić, M., Todosijević, R., & Urošević, D. (2019). Less is more: General variable neighborhood search for the capacitated modular hub location problem. *Computers & Operations Research*, *110*, 101–115.

Nekooghadirli, N., Tavakkoli-Moghaddam, R., Ghezavati, V., & 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.

Poonthalir, G., & Nadarajan, R. (2018). A Fuel Efficient Green Vehicle Routing Problem with varying speed constraint (F-GVRP). *Expert Systems with Applications*, *100*, 131–144.

Rafie-Majd, Z., Pasandideh, S., & 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.

Rayat, F., Musavi, M., & Bozorgi-Amiri, A. (2017). Bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks: A modified AMOSA approach. *Applied Soft Computing*, *59*, 622–643.

Saif-Eddine, A., El-Beheiry, M., & El-Kharbotly, A. (2019). An improved genetic algorithm for optimizing total supply chain cost in inventory location routing problem. *Ain Shams Engineering Journal*, *10*, 63–76.

Saragih, N., Bahagia, S., Suprayogi, & Syabri, I. (2019). A heuristic method for location-inventory-routing problem in a three-echelon supply chain system. *Computers & Industrial Engineering*, *127*, 875–886.

Seyedhosseini, S., Bozorgi-Amiri, A., & Darei, S. (2014). An integrated location-routing-inventory problem by considering supply disruption. *iBusiness*, *6*, 29–37.

Simeonova, L., Wassan, N., Salhi, S., & Nagy, G. (2018). The heterogeneous fleet vehicle routing problem with light loads and overtime: Formulation and population variable neighborhood search with adaptive memory. *Expert Systems with Applications*, *114*, 183–195.

Skouri, K., Sifaleras, A., & Konstantaras, I. (2018). Open problems in green supply chain modeling and optimization with carbon emission targets. In P. M. Pardalos, & A. Migdalas (Eds.), *Open Problems in Optimization and Data Analysis* (pp. 83–90). Springer Optimization and Its Applications.

Soon, K., Lim, J., Parthiban, R., & M.C., H. (2019). Proactive eco-friendly pheromone-based green vehicle routing for multi-agent systems. *Expert Systems With Applications*, *121*, 324–337.

Tavakkoli-Moghaddam, R., Forouzanfar, F., & 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*, *9*. doi:10.1186/2251-712X-9-19.

Tavana, M., Abtahi, A., Caprio, D., Hashemi, R., & Yousefi-Zenouz, R. (2018). An integrated location-inventory-routing humanitarian supply chain network with pre- and post-disaster management considerations. *Socio-Economic Planning Sciences*, *64*, 21–37.

Todosijević, R., Mladenović, M., Hanafi, S., Mladenović, N., & Crévits, I. (2016). Adaptive general variable neighborhood search heuristics for solving the unit commitment problem. *International Journal of Electrical Power & Energy Systems*, *78*, 873–883.

Toro, M., Franco, F., Echeverri, M., & Guimaraes, F. (2017). A multi-objective model for the green capacitated location-routing problem considering environmental impact. *Computers & Industrial Engineering*, *110*, 114–125.

Vahdani, B., Veysmoradi, D., Noori, F., & Mansour, F. (2018). Two-stage multi-objective location-routing-inventory model for humanitarian logistics network design under uncertainty. *International Journal of Disaster Risk Reduction*, *27*, 290–306.

Wang, S., & Ye, B. (2018). A comparison between just-in-time and economic order quantity models with carbon emissions. *Journal of Cleaner Production*, *187*, 662–671.

Xu, M., Cui, Y., Hu, M., Xu, X., Zhang, Z., Liang, S., & Qu, S. (2019). Supply chain sustainability risk and assessment. *Journal of Cleaner Production*, *225*, 857–867.

Xu, Z., & Cai, Y. (2018). Variable neighborhood search for consistent vehicle routing problem. *Expert Systems with Applications*, *113*, 66–76.

Yu, Y., Wang, S., Wang, J., & Huang, M. (2019). A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows. *Transportation Research Part B: Methodological*, *122*, 511–527.

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

Zhang, B., Li, H., Li, S., & Peng, J. (2018). Sustainable multi-depot emergency facilities location-routing problem with uncertain information. *Applied Mathematics and Computation*, *333*, 506–520.

Zhang, S., Lee, C., Wu, K., & Choy, K. (2016). Multi-objective optimization for sustainable supply chain network design considering multiple distribution channels. *Expert Systems with Applications*, *65*, 87–99.

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

Zheng, X., Yin, M., & Zhang, Y. (2019). Integrated optimization of location, inventory and routing in supply chain network design. *Transportation Research Part B: Methodological*, *121*, 1–20.