Green vehicle routing and scheduling problem with heterogeneous fleet including reverse logistics in the form of collecting returned goods

Article *in* Applied Soft Computing · September 2020

DOI: 10.1016/j.asoc.2020.106462

Reza Alizadeh Foroutan
Mazandaran University of Science and Technology

Javad Rezaeian Zeidi
Mazandaran University of Science and Technology

Iraj Mahdavi
Mazandaran University of Science and Technology

Green vehicle routing and scheduling problem with heterogeneous fleet including reverse logistics in the form of collecting returned goods

Reza Alizadeh Foroutan[1] , Javad Rezaeian[1,*], Iraj Mahdavi[1]

[1]*Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran*

***Abstract*— Vehicle routing problem (VRP) is about finding optimal routes for a fixed fleet of vehicles in order that they can meet the demands for a set of given customers by traveling through those paths. This problem and its numerous expansions are one of the most important and most applicable transportation and logistics problems. In this study, the green vehicle routing and scheduling problem with heterogeneous fleet including reverse logistics in the form of collecting returned goods along with weighted earliness and tardiness costs is studied to establish a trade-off between operational and environmental costs and to minimize both simultaneously. In this regard, a mixed integer non-linear programming (MINLP) model is proposed. Since the problem is categorized as NP-hard, two meta-heuristics, a simulated annealing (SA) and a genetic algorithm (GA) are suggested in order to find near-optimal solutions for large instances in a reasonable computational time. The performances of the proposed algorithms are evaluated in comparison with the mathematical model for small-sized problems and with each other for problems of all size using a set of defined test problems. Analysis of the results considering two criteria: solutions quality and computational times, indicates the satisfactory performance of the presented algorithms in a proper computational time. Meanwhile, a statistical hypothesis testing (T-test) is conducted. It can generally be observed that SA achieves relatively better results in terms of solution quality, while GA spends less computational time for all-sized test problems. Eventually, sensitivity analysis is conducted to investigate the effect of collecting returned goods on the cost of total CO_2 emissions, variable costs of the fleet and the objective function value.**

***Keywords*— Green vehicle routing and scheduling, Heterogeneous fleet, Reverse logistics, Time-dependent traffic patterns, CO_2 emissions**

1. Introduction and literature review

Today, transportation problems are one of the most important and most practical problems in the field of logistics and supply chain management. In this regard, designing product and service distribution networks as one of the most significant stages of this scope has always been interested by many researchers, manufacturers and service providers. Vehicle routing problem (VRP) is one of the most important, applicable and meanwhile most common transportation problems. It was proved that VRP is NP-Hard [1]. The classical form of VRP is depicted in Fig. 1 in the following.

* Corresponding Author (Email): Javad Rezaeian (j.rezaeian@ustmb.ac.ir)

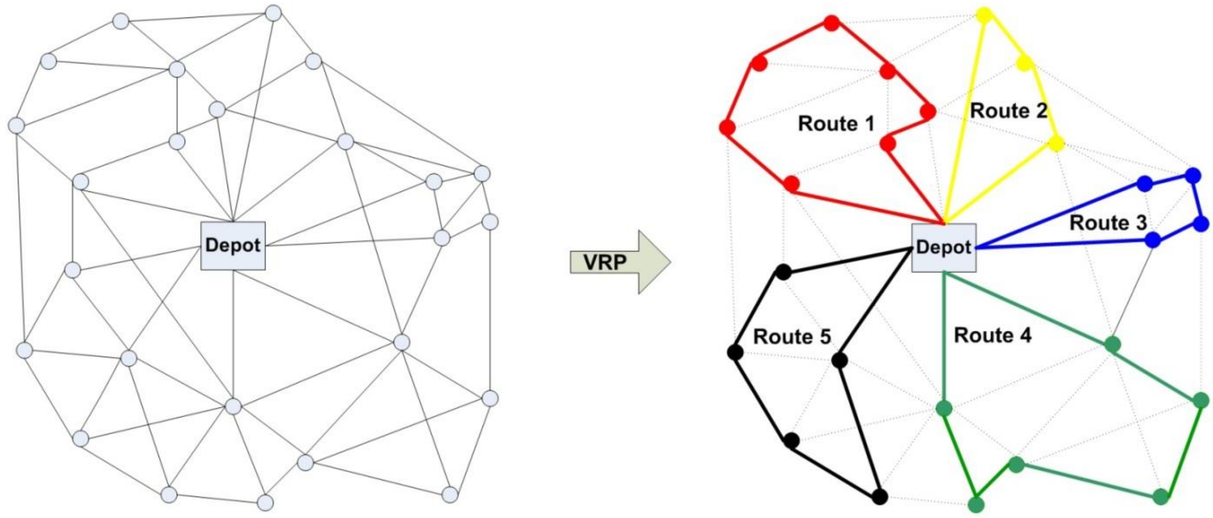

Fig. 1 A representation of the classical VRP

Transportation costs significantly affect final prices of products and consequently the level of customers' satisfaction. Hence, VRP as one of the most practical methods of reducing transportation costs has always been a concern for supply chain management researchers in recent years. On the other hand, green logistics and reverse logistics (RL) as a particular form of logistics are increasingly considered by governments, industry owners and scholars nowadays, since it can lead to sustainable development, maintaining environmental standards and the stability of supply and distribution strategies. Collecting returned goods in the form of after-sales services can be mentioned as one of the most known types of RL. This procedure is the RL approach applied in this paper. It directly causes income rises for manufacturers and by increasing customer's satisfaction can results in further potential revenues. In addition, collecting returned goods and restoring them back to the product cycle instead of just considering them as wastes would be a proper practical approach to deal with environmental concerns.

Vehicle routing problem was first introduced by Dantzig and Ramser as a generalized form of traveling salesman problem (TSP) [2]. Traditionally, the vehicle routing problem pursues the idea of minimizing transportation costs of raw materials, final products or service deliveries between the producer and consumer levels in a supply chain. Over the last two decades, environmental concerns have led to a new category in the VRP called "Green VRP" which is trying to minimize the environmental costs along with VRP's common operational costs. Greenhouse gases emissions, especially carbon dioxide (CO_2) - 90% of the entire emissions – are the main cause of the global warming phenomenon during recent decades. According to the international energy agency, transportation, after electricity and heat generation was the second largest CO_2 producing sector worldwide in 2013 (23% of the whole CO_2 emissions) [3]. Whereas three quarters of the universal CO_2 emissions is caused by the road transportation sector, any kind of scientific or practical attempt to decrease those emissions seems necessary more than ever.

Among the first efforts in this context, a paper by Bloemhof-Ruwaard et al. [4] in 1995 as well as Daniel et al. [5] in 1997 can be mentioned. They discussed the interaction between operations research-based methods and environmental management. Green vehicle routing problem with the aim of creating a trade-off between operational and environmental costs has become the topic of researchers' focus since 2006 and has always been the subject of attention in recent years. Fig. 2 displays some of the most important extensions on vehicle routing problem. It shows the appearance time of the green vehicle routing problem in the timeline of VRPs.

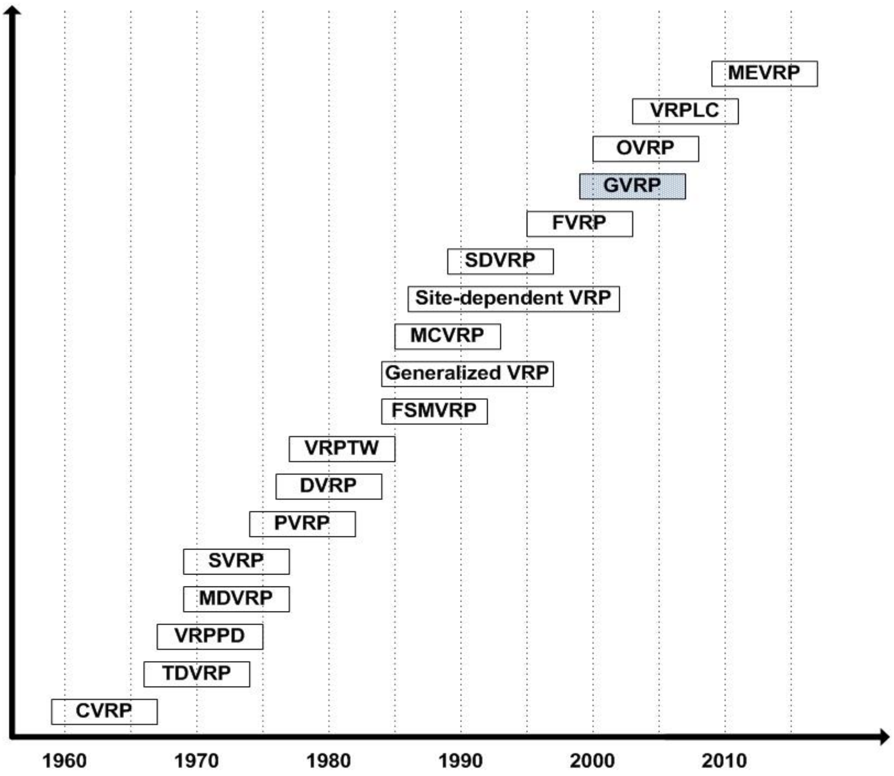

Fig. 2 The timeline of vehicle routing problems [1]

The green vehicle routing problem has been studied by many researchers during the last decade [6-8]. This problem was classified in three major categories by Lin et al. [9] in their extensive review of the literature: (i) G-VRP (green vehicle routing problem), (ii) VRPRL (VRP in reverse logistics) and (iii) PRP (pollution routing problem).

Mirzapour and Rekik formulated a multi-product and multi-period inventory routing problem in the format of a mixed integer linear model with the intention of increasing the performance of supply chain. They made the transshipment option available to show its environmental impacts [10]. Coelho et al. developed a meta-heuristic inspired by neighborhood search method for a VRP with pickup and delivery (VRPPD) with selective pickups in order to achieve a sustainable green supply chain [11]. Soysal proposed a probabilistic mixed-integer linear programming for a closed-loop inventory routing problem with forward and reverse logistics and including demand uncertainty and multiple products. He pursued the goal of minimizing a number of operational costs and the environmental cost of energy consumption [12].

The pollution routing problem has been widely studied in recent years. Franceschetti et al. offered time-dependent PRP (TDPRP) combining PRP and TDVRP [13]. Kwon et al. introduced a development on heterogeneous fixed fleet vehicle routing

1. CVRP: Capacitated VRP; TDVRP: Time Dependent VRP; VRPPD: VRP with Pickup and Delivery; MDVRP: Multi-depot VRP; SVRP: Stochastic VRP; PVRP: Periodic VRP; DVRP: Dynamic VRP; VRPTW: VRP with Time Windows; FSMVRP: Fleet Size and Mix VRP; MCVRP: Multi-compartment VRP; SDVRP: Split-delivery VRP; FVRP: Fuzzy VRP; GVRP: Green VRP; OVRP: Open VRP; VRPLC: VRP with Loading Constraints; MEVRP: Multi-echelon VRP.

problem (HFFVRP) considering carbon emissions trading cost [14]. Bektas and Laporte represented a PRP with the aim of minimizing total cost of fuel consumption and greenhouse gas emissions with the cost of drivers' wages at the same time [15]. Demir along with Bektas and Laporte developed the previous mentioned problem by adding speed restriction [16]. They also proposed a further development on PRP intending to minimize fuel consumption and total time traveled on the entire routes [17]. Koç et al. represented a hybrid evolutionary algorithm to minimize fuel consumption costs, CO_2 emissions, total drivers' wages and total fixed costs for a heterogeneous fleet [18]. Figliozzi minimized total distance traveled and also total spent times on roads in addition to the cost of vehicles and cost of emissions [19].

One of the latest problems in the literature is the green vehicle routing and scheduling problem (GVRSP) which was introduced by Xiao and Konak in 2015 [20]. They proposed a multi-objective function with the goal of minimizing CO_2 emissions and solved it with a hierarchical approach. Whenever the travel time between two nodes is depended on the distance between them and the time of the day, VRP is expressed as time dependent VRP (TDVRP) [21]. A common assumption in most of TDVRP studies is that each vehicle is allowed to travel the distance between two nodes in only one time period, while in practice for many reasons such as permitted stops, traffic congestion and so on, the vehicle might need more than a single time period to travel that distance. The GVRSP eliminated this deficiency assuming that the travel between any two nodes is allowed in more than a unit time period. In the same way, in this study the scheduling horizon is divided into multiple time periods. Since the travel speed of vehicles in the case of urban and occasionally suburban roads varies due to the road conditions, traffic congestion and the time of the day, it is assumed that the travel speed of vehicles is constant for each arc in each time period and it is considered equal to the average speed during that period. For each arc, one of the time-dependent traffic patterns is applied depending on the nature of its traffic congestion. These patterns which are divided into 6 types are shown in Fig. 3.

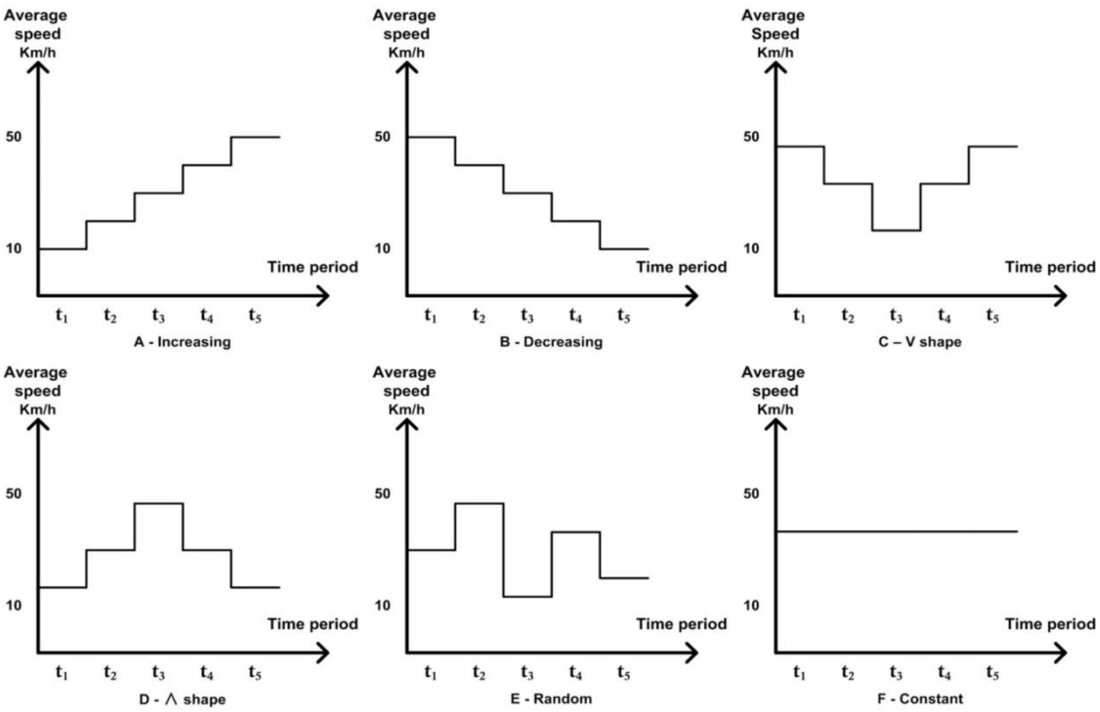

Fig. 3 Time-dependent speed patterns

Xiao and Konak expanded the GVRSP and represented a new MILP model considering heterogeneous fleet and the impact of vehicle loads on emissions. They aimed to minimize CO_2 emissions using a hybrid algorithm of exact and heuristic methods [30]. They studied a time-dependent GVRSP with CO_2 emissions optimization and developed an exact dynamic programming algorithm to specify the optimal vehicle schedules proposing a hybrid solution approach combined of a genetic algorithm and the exact dynamic programming procedure [31]. Raeesi and Zografos formulated the GVRSP as a bi-objective time-and-load-dependent optimization model and proposed an algorithm to solve it on a real road network [32].

Zhang et al. proposed a joint optimization model of the GVRSP with time-varying speeds considering extra wages during nonworking periods and soft time-window constraints. To deal with the model, they presented a heuristic algorithm based on the adaptive large neighborhood search. Eventually, a numerical simulation example is prepared to demonstrate the optimization model and the algorithm [33]. Musavi et al. presented a bi-objective model for a green truck scheduling and routing problem at a cross-docking system considering a learning effect in loading and unloading process times trying to minimize time window violations and total fuel consumption of trucks. An archived multi-objective simulated annealing (AMOSA) is applied and modified to solve the proposed model [34]. Nabil et al. proposed a mixed integer programming model for the GVRSP with time dependent travel speeds and heterogeneous fleet with the aim of minimizing CO_2 emissions by minimizing the deviation of travel speed from optimum travel speeds [35]. Androutsopoulos and Zografos formulated and solved a bi-objective time, load and path-dependent VRP with time windows (BTL-VRPTW) incorporating cost and pollution criteria. The objective was to determine minimum travel time and fuel consumption. A k-shortest path approach is proposed to deal with the computational burden of the problem and a two-stage generic solution framework is implemented by integrating the routing and scheduling method into an Ant Colony System algorithm to solve the problem. They also applied a label setting algorithm to transform VRP routes to non-dominated solutions [36]. Rabbani et al. proposed a new multi-objective model for VRP under a stochastic uncertainty including resiliency factors. They tried to minimize the total transportation cost, traffic pollution, customer dissatisfaction and maximize the reliability of vehicles. They developed a Simulated Annealing algorithm for solving the proposed model [37]. Finally, Matos et al. presented a mathematical formulation and a hybrid algorithm that combines the meta-heuristic Iterated Local Search (ILS), the Random Variable Neighborhood Descent (RVND) procedure and an exact Set Covering model to deal with GVRSP aiming to minimize carbon dioxide emissions [38].

Since collecting returned goods (such as defective, damaged, expired, etc.) makes it possible to recycle and send them back into consumption cycle or destroy them with the least environmental pollutions, it is an effective way of contributing to environmental protection and at the same time obtaining economic benefits. Therefore, collecting returned goods is continuously becoming more common among various industries and also environmentally friendly organizations undertake parts of costs for both scientific researches and practical activities in this regard. Hence, considering green concerns in VRP, especially collecting returned goods, which has not been addressed in the literature of GVRSP so far is the main motivation of the present study. Many industries face with the features described in this study. Among many of the real-world cases, home appliance and food industries are the well-known examples in which products that are damaged during the distribution and delivery process can either be disposed of or returned to the recycling cycle, especially the ones with plastic packaging or those partially made of glass.

Thus, the GVRSP is developed in order to establish a trade-off between the operational and environmental costs. To do so, three ideas are pursued: 1) reducing total CO_2 emissions of the vehicles as the major greenhouse gas and the main cause of the global warming phenomenon; 2) collecting returned goods within the framework of reverse logistics which can turn an environmental threat into an opportunity and make economic revenue by recycling goods in addition to helping to preserve the environment; 3) decreasing fuel consumption cost as a part of variable costs of the fleet. Fixed and variable operational costs are calculated based on the number and type of selected vehicles that form the tours and also the amount of their cargo on the road (whether new products or returned goods). The gross weight of each vehicle may be changing continuously on its path from node to node, because of demands delivery and collecting returned goods. It affects both total CO_2 emissions and fuel consumptions. This feature also has not been taken into consideration in GVRSP and therefore is another novelty of this study.

In today's competitive world, almost all customers expect their demands to be fulfilled at the appropriate time before their due date. Hence, usually in cases where customers' demands are met after due date, a tardiness penalty will be imposed that is different proportional to the importance of each customer. Also, since the earliness on demand delivery for various reasons such as storage costs or human resources allocations can lead to customer dissatisfaction, an earliness penalty proportional to the weight of each customer can be considered for the orders that are delivered ahead of due times. Hence, in addition to tardiness penalty, we considered earliness penalty in this paper as well.

In this study, the green vehicle routing and scheduling problem with heterogeneous fleet including reverse logistics in the form of collecting returned goods along with weighted earliness and tardiness costs is investigated which is described as follows: A fleet of heterogeneous vehicles is deployed in a central depot. They are supposed to completely meet the demands of a set of customers and meanwhile collect the proper amount of their returned goods. The problem is formulated using the network $G(\mathcal{N}, \boldsymbol{\mathcal{A}})$ in which $\mathcal{N}$ represents a set of customer nodes plus the central depot node and $\boldsymbol{\mathcal{A}}$ is the set of arcs or routes between nodes. Inspiring by local companies, we considered nodes as distribution centers of a supply chain located in different cities in a county, province or state with real distances. Each customer has a certain demand, a determined amount of potential returned goods and also a pre-defined due date so that any earliness or tardiness on delivering its demands will impose a weighted penalty to the objective function based on its importance. Also, to calculate the speed of vehicles at any time on different routes, time-dependent traffic patterns are applied.

The rest of the paper is organized as follows: in the next section, the proposed MINLP model along with its parameters and variables and the methods which are applied to calculate CO_2 emissions and fuel consumption are introduced in detail. Then in section (3), suggested model is validated by solving a small-sized instance in a three-stage procedure using LINGO 9.0. After that in section (4), a simulated annealing and a genetic algorithm are proposed in order to make the model applicable to the real-world implementations. Section (5) is about design of experiments and parameters setting where the performances of the presented meta-heuristics in terms of solutions quality and computational times are evaluated using several numerical examples and T-tests. Also, sensitivity analysis is performed to determine the effect of collecting returned goods on some model outcomes. At last in section (6), conclusions and proposed ideas for future researches are given.

2. Problem definition

In this section, problem assumptions, indices, parameters and decision variables are introduced; then the proposed mathematical model as a mixed integer non-linear programming problem is presented and at last the applied methods to calculate the amount of CO_2 emissions and fuel consumption are described in detail. The fleet consists of p different vehicle types ($l = 1, 2, \ldots, p$) where the number (NV_l) and the capacity of each type (C_l) are given. The set of customers comprises n nodes ($\mathcal{N}' = \{1, 2, \ldots, n\}$) and the scheduling horizon is divided into m time periods ($k = 1, 2, \ldots, m$).

2.1. Assumptions

- The fleet is made of heterogeneous vehicles.
- The model is single-product and demands are deterministic.
- Delivery of products to customers is carried out only by existing vehicles.
- Every customer demand has to be fulfilled in a single meeting by only one vehicle.
- None of the orders can violate the capacity of all types of vehicles. In other words, for each $1 \leq i \leq n$ there must be at least one l such that $Q_i \leq C_l$.
- The total demands of all customers cannot exceed the total capacity of existing vehicles.
- No vehicle is permitted to load up more than its capacity on its tour.
- Each vehicle must start its tour from the depot and finish it back there too.
- Vehicles are allowed to travel any arc in more than one time period.
- The speed of a vehicle on each specific arc depends on the time period it travels that arc.
- The speed of every vehicle at any particular time period on each specific arc is constant and equals to the average traveling speed in that time period on that arc.
- The total weight of each vehicle is variable during its tour depending on the total amount of new and returned products shipped by it.
- Any earliness or tardiness in customer service is penalized according to customers' weights.

2.2. Notations:

Indices

i	index of nodes ($i = 0, 1, \ldots, n$), the depot is denoted by $i = 0$
l	index of vehicles type ($l = 1, \ldots, p$)
k	index of time periods ($k = 1, \ldots, m$)

Parameters

$\mathcal{N}$	set of all nodes including the depot, $i \in \mathcal{N}$; ($i = 0$ is the depot)
$\boldsymbol{\mathcal{A}}$	set of arcs; each arc is composed of two nodes: $(i, j) \in \boldsymbol{\mathcal{A}}$, $\forall\, i, j \in \mathcal{N},\ i \neq j$
NV_l	number of vehicles type l
C_l	capacity of vehicle type l
W_l	weight of vehicle type l
Q_i	demand of customer i (number of demanded goods)
D_{ij}	distance of arc (i, j)
K	set of time periods, $k \in K$
P	set of vehicles types, $p \in P$
ε_{ij}^{lk}	CO_2 emissions rate per unit distance traveled on arc (i, j) by vehicle type l in period k; (g/km)

LC_{ij}^{lk}	load correction factor for vehicle type l when travels arc (i,j) in period k
φ_i	tardiness penalty coefficient for customer i
δ_i	earliness penalty coefficient for customer i
α_l	variable cost of traveling unit distance by vehicle type l
c	cost of unit fuel consumption
ω	cost of unit CO_2 emissions
F_l	fixed cost of vehicle type l
LD	the minimum required distance to traverse for any arc which is selected to be traveled
v_{ij}^k	vehicles constant velocity on arc (i,j) in period k
ξ_{1i}	unload time of a unit product from a vehicle in node i
ξ_{2i}	load up time of a unit product on a vehicle in node i
b_k	start time of time period k
e_k	finish time of time period k
DT_i	Due date of customer i
β	the revenue gained from the collecting of a single returned product
R_i	the number of potential returned products in node i
w	weight of a single product
M	a large number

Decision variables

X_{ij}^l	if arc (i,j) is selected to be traveled by vehicle type l is 1, otherwise 0
x_{ij}^{lk}	if arc (i,j) is selected to be traveled by vehicle type l in time period k is 1, otherwise 0
y_{ij}^l	the number of new products vehicle type l is carrying while traveling arc (i,j)
r_{ij}^l	the number of returned products vehicle type l is carrying while traveling arc (i,j)
BRY_{ij}^l	equals 1 if r_{ij}^l is greater than zero, otherwise 0
d_{ij}^{lk}	traveled distance of arc (i,j) if it is traversed by vehicle type l in time period k
t_{ij}^{lk}	travel time of vehicle type l on arc (i,j) in time period k
l_i	vehicle departure time from node i
a_i	vehicle arrival time in node i

T_i	tardiness of customer i
E_i	earliness of customer i
ρ_{ij}^{l}	fuel consumption rate of vehicle type l per unit distance traveled on arc (i,j)
y_i	number of collected returned products from node i

2.3. The Mathematical Model

The proposed MINLP model is formulated as follows:

$$\min Z = \sum_{i=0}^{n}\sum_{l=1}^{p}\sum_{k=1}^{m}(\varepsilon_{i0}^{lk}.\left(1 - BRY_{i0}^{l}\right).d_{i0}^{lk} + \varepsilon_{i0}^{lk}.BRY_{i0}^{l}.LC_{i0}^{lk}.d_{i0}^{lk} + \sum_{j=1}^{n}\varepsilon_{ij}^{lk}.\ LC_{ij}^{lk}.d_{ij}^{lk})\ .\omega$$
$$+\sum_{l=1}^{p}\sum_{j=1}^{n}F_l\ .\ X_{0j}^{l} + \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{l=1}^{p}\sum_{k=1}^{m}(\alpha_l + c.\rho_{ij}^{l}).d_{ij}^{lk} - \beta.\sum_{i=0}^{n}y_i +$$
$$\sum_{i=1}^{n}\varphi_i\ .T_i + \sum_{i=1}^{n}\delta_i\ .E_i \tag{1}$$

Subject to:

$$\sum_{l=1}^{p}\sum_{i=0}^{n}X_{ij}^{l} = 1 \qquad \forall j \in \mathcal{N}|\{0\} \tag{2}$$

$$\sum_{l=1}^{p}\sum_{j=0}^{n}X_{ij}^{l} = 1 \qquad \forall i \in \mathcal{N}|\{0\} \tag{3}$$

$$\sum_{i=0}^{n}X_{ij}^{l} - \sum_{i=0}^{n}X_{ji}^{l} = 0 \qquad \forall j \in \mathcal{N}\ |\{0\}, \forall l \in P \tag{4}$$

$$\sum_{j=1}^{n}X_{0j}^{l} \le NV_l \qquad \forall l \in P \tag{5}$$

$$\sum_{l=1}^{p}\sum_{i=0}^{n}y_{ij}^{l} - \sum_{l=1}^{p}\sum_{i=0}^{n}y_{ji}^{l} = Q_j \qquad \forall j \in \mathcal{N}|\{0\} \tag{6}$$

$$Q_j\ .X_{ij}^{l} \le y_{ij}^{l} \le (C_l - Q_i).X_{ij}^{l} \qquad \forall i,j \in \mathcal{N}, i \ne j, \forall l \in P \tag{7}$$

$$\sum_{l=1}^{p}\sum_{i=0}^{n}r_{ji}^{l} - \sum_{l=1}^{p}\sum_{i=0}^{n}r_{ij}^{l} = y_j \qquad \forall j \in \mathcal{N}\ |\{0\}, \forall l \in P \tag{8}$$

$$r_{ij}^l \leq (C_l - Q_j).X_{ij}^l \qquad \forall i,j \in \mathcal{N}, i \neq j, \forall l \in P \tag{9}$$

$$y_{ij}^l + r_{ij}^l \leq C_l \qquad \forall i,j \in \mathcal{N}, i \neq j, \forall l \in P \tag{10}$$

$$y_i \leq R_i \qquad \forall i \in \mathcal{N} \tag{11}$$

$$X_{ij}^l \geq x_{ij}^{lk} \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P, \forall k \in K \tag{12}$$

$$X_{ij}^l \leq \sum_{k=1}^{m} x_{ij}^{lk} \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P \tag{13}$$

$$D_{ij}.X_{ij}^l = \sum_{k=1}^{m} d_{ij}^{lk} \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P \tag{14}$$

$$LD.x_{ij}^{lk} \leq d_{ij}^{lk} \leq D_{ij}.x_{ij}^{lk} \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P, \forall k \in K \tag{15}$$

$$t_{ij}^{lk} = d_{ij}^{lk}/v_{ij}^k \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P, \forall k \in K \tag{16}$$

$$l_i \leq e_k.x_{ij}^{lk} - t_{ij}^{lk} + e_m.(1 - x_{ij}^{lk}) \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P, \forall k \in K \tag{17}$$

$$a_j \geq b_k.x_{ij}^{lk} + t_{ij}^{lk} - e_m.(1 - x_{ij}^{lk}) \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P, \forall k \in K \tag{18}$$

$$a_i + \xi_{1i}.Q_i + \xi_{2i}.y_i \leq l_i \qquad \forall i \in \mathcal{N}|\{0\} \tag{19}$$

$$a_0 \leq e_m \tag{20}$$

$$a_j \geq l_i + \sum_{k=1}^{m} t_{ij}^{lk} - e_m.(1 - X_{ij}^l) \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P \tag{21}$$

$$t_{ij}^{lk} \leq e_k - b_k \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P, \forall k \in K \tag{22}$$

$$x_{jj'}^{lk} \leq 2 - x_{ij}^{lk'} - X_{ij}^l \qquad \forall i,j,j' \in \mathcal{N}|\{0\}, \forall l \in P, \forall k,k' \in K; k < k' \tag{23}$$

$$T_i = max(a_i - DT_i, 0) \qquad \forall i \in \mathcal{N}|\{0\} \tag{24}$$

$$E_i = max(DT_i - a_i, 0) \qquad \forall i \in \mathcal{N}|\{0\} \tag{25}$$

$$r_{ij}^l \geq BRY_{i0}^l \qquad \forall i \in \mathcal{N}, j = 0, \forall l \in P \tag{26}$$

$$X_{ij}^l \geq BRY_{i0}^l \qquad \forall i \in \mathcal{N}, j = 0, \forall l \in P \tag{27}$$

$$r_{ij}^l.X_{ij}^l \leq BRY_{ij}^l.M \qquad \forall i \in \mathcal{N}, j = 0, \forall l \in P \tag{28}$$

$$X_{ij}^l, x_{ij}^{lk}, BRY_{i0}^l \in \{0,1\} \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, l \in P, k \in K \tag{29}$$

$$y_{ij}^l, r_{ij}^l, d_{ij}^{lk}, t_{ij}^{lk}, l_i, a_i, T_i, E_i, \rho_{ij}^l, y_i \geq 0 \qquad \forall i,j \in \mathcal{N}, l \in P, k \in K \tag{30}$$

The proposed objective function (1) minimizes both operational and environmental costs at the same time. The first term of this function minimizes the cost of total CO_2 emissions. The second term calculates fixed cost of the fleet. The third phrase is used to optimize the operational and environmental costs simultaneously by minimizing variable costs of the fleet including fuel consumption cost. The fourth expression calculates the total revenue gained by collecting returned goods. And finally, the fifth and sixth terms try to minimize total tardiness and earliness respectively in order to deliver customers' demands on time and consequently reduce operational costs caused by late or early delivered orders. Constraints (2) and (3) guarantee every node except for the depot ($i = 0$) to have only one entry and one exit respectively. Constraint (4) states each vehicle which enters node j , ($j \neq 0$) must also leave it. This constraint actually ensures continuity of routes. Constraint (5) implies that the number of vehicles of any kind is limited. Constraint (6) guarantees each customer demand must be fulfilled completely. Constraint (7) ensures that the limited capacity of each vehicle for shipping new products must be met. Constraint (8) determines the number of returned goods to be collecting from each customer. Constraint (9) guarantees that the shipping capacity of returned products by each vehicle would not be violated. Constraint (10) ensures that a cargo (new products plus returned ones) carried by each vehicle on each arc is not allowed to exceed vehicle capacity. Constraint (11) defines the upper bound of collecting returned goods for each customer. Constraint (12) states that, in order to travel path (i, j) by vehicle type l in time period k, this arc has to be selected for traveling by vehicle type l. Constraint (13) explains if path (i, j) is selected to be traveled by vehicle type l, this arc can be traveled in at least one and at most K time period. The two recent restrictions together make variables X_{ij}^{l} and x_{ij}^{lk} consistent with each other. Constraint (14) guarantees that if an arc is selected to be traveled by any vehicle, the path must be traveled to the end. Constraint (15) makes variable d_{ij}^{lk} consistent with variable x_{ij}^{lk}. It also determines the least and the most can-be-traveled distance when path (i, j) is selected to be traveled by vehicle type l in time period k. Constraint (16) defines travel time of vehicle type l in time period k on path (i, j). Constraints (17) and (18) respectively determine departure and arrival times of vehicles from/in nodes in any time periods. Constraint (19) makes sure that at each node, departure time should be occurred after arrival time plus service time. Service time at each node is equal to the total time of loading and unloading goods. Constraint (20) states that the returning time to the depot cannot exceed the completion time of the last time period. Constraint (21) calculates the arrival time at node j. The last three constraints together prevent sub-tours to be created. Constraint (22) ensures that no overuse of the time periods would happen which means the total travel time of each single vehicle in each time period should be less than the duration of the time period. Constraint (23) indicates that if vehicle type l in time period k' travels arc (i, j), it cannot be able to travel arcs that start from node j in time period k such that $k < k'$. Constraints (24) and (25) respectively are used to compute the tardiness and earliness of demand delivery to each customer. Three constraints (26-28) together calculate binary variable BRY_{i0}^{l}. With regard to the equation used to calculate CO_2 emissions in this paper, one of the contributing factors is whether a moving vehicle is loaded or not. Hence, by considering the problem assumptions, load absence in a moving vehicle is only possible on the last arc of each tour (the arc leading to the depot). Thus, in order to calculate the correct amount of gas emissions and its corresponding costs, defining the mentioned binary variable to distinguish load presence or absence in vehicles moving on the arcs that lead to the depot is essential. And eventually constraints (29) and (30) describe the range of decision variables.

It should be noted that constraints (2-11) are used to make planning-related decisions including order fulfillments and collecting the optimal amounts of returned goods. Constraints (12-25) are applied to make scheduling-related decisions in a continuous time horizon. Constraints (26-28) are exclusively defined to calculate the environmental costs and constraints (29) and (30) specify the range of decision variables.

2.4. Calculation of CO_2 emissions

In order to reduce CO_2 emissions produced by road transportations, as the most important air pollutant greenhouse gas, lots of studies have been done in recent years. There are two common known methods to calculate CO_2 emissions rate; one is using the Motor Vehicle Emission Simulator (MOVES) which is an emissions modeling system that estimates emissions for mobile sources presented by United States Environmental Protection Agency (U.S. EPA) and another is applying existing emission models from the literature. At the moment, there are several emissions and energy consumption models that differ with each other in terms of modeling techniques, model structure and data requirements. Demir et al. [22] reviewed some of these models and categorized them in three main classes according to their level of complexity as follows: (i) Factor models that include simple methods of fuel consumption; (ii) Macroscopic models that are used to estimate emission rates at the network level and (iii) Microscopic models which are used to estimate instantaneous fuel consumption and gas emission rates at a more detailed level.

One of the methods that have been used frequently in the literature is the methodology for calculating transportation emissions and energy consumption or so-called MEET which is introduced and developed by Hickman et al. [23] for heavy-duty vehicles. This method which is categorized as a macroscopic model in Demir's classification is applied to calculate CO_2 emissions in this paper. MEET is a method based on on-road measurements and all of its parameters have been derived from actual experiences. According to Hickman factors such as vehicle technology, average speed, vehicle mileage, engine temperature and road height directly affect fuel consumption and subsequently have influence on emissions rates. Also, additional factors such as vehicle's load, its gross weight, road gradient, ambient temperature and humidity percentage that make changes in engine function, indirectly affect emission rates. The introduced regression function in this method is:

$$\varepsilon = \mathrm{K} + av + bv^2 + cv^3 + \frac{d}{v} + \frac{e}{v^2} + \frac{f}{v^3} \tag{31}$$

where ε is the rate of emission in g/km for an unloaded goods vehicle, or for a bus or coach carrying a mean load, on a road with a gradient of 0%. K is a constant, $a - f$ are coefficients and v is the mean velocity of the vehicle in km/h. They were derived by Hickman for four classes of heavy goods vehicle (3.5 to 7.5 tones, 7.5 to 16 tones, 16 to 32 tones and 32 to 40 tones) as shown in Table 1. Also, the relationship between the amount of CO_2 emission rates and vehicle's speeds are portrayed in Fig. 4.

Table 1. Coefficients of CO_2 emission for heavy goods vehicles

Gross weight (tones)	**K**	***a***	***b***	***c***	***d***	***e***	***f***
3.5-7.5	110	0	0	0.000375	8702	0	0
7.5-16	871	-16	0.143	0	0	32031	0
16-32	765	-7.04	0	0.000632	8334	0	0
32-40	1576	-17.6	0	0.000117	0	36067	0

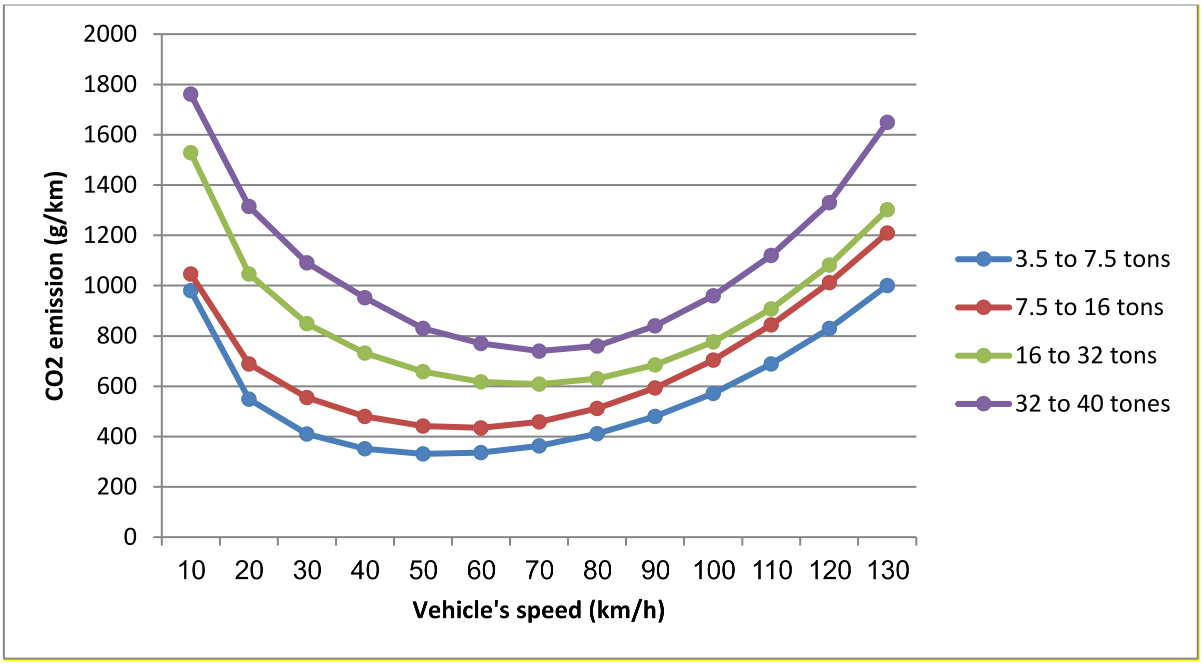

Fig. 4 relationship between the amount of CO_2 emission rates and vehicle's speeds for heavy-duty vehicles

Since in the case of heavy-duty vehicles, the vehicle load has a significant influence on fuel consumption and CO_2 emissions, therefore the calculated emission rate via equation (31) is corrected considering road gradient, vehicle's load and traveled distance as follows:

$$F = \varepsilon \; . \; GC \; . LC \; . \; D \qquad (32)$$

Where F is the corrected emission rate for a loaded vehicle, GC is the gradient correction factor, LC is the load correction factor and D is the traveled distance. Since in this study, the road gradient is considered 0% (standard conditions), GC in above equation takes the value 1 $(GC = 1)$. The load correction factor (LC) is calculated as below:

$$LC = \kappa + n\gamma + p\gamma^2 + q\gamma^3 + rv + sv^2 + tv^3 + \frac{u}{v} \qquad (33)$$

In the above equation κ is a constant, $n - u$ are coefficients, γ is the gradient in percent and v is the mean velocity of the vehicle in km/h. The coefficients of different weight classes of heavy-duty vehicles are listed in Table 2.

Table 2. Coefficients of load correction equation

Gross weight (tones)	κ	n	p	q	r	s	t	u
3.5-7.5	1.27	0.0614	0	-0.00110	-0.00235	0	0	-1.33
7.5-16	1.26	0.0790	0	-0.00109	0	0	-2.03×10^-7	-1.14
16-32	1.27	0.0882	0	-0.00101	0	0	0	-0.483
32-40	1.43	0.121	0	-0.00125	0	0	0	-0.916

2.5. Calculation of fuel consumption

According to the report published by Ministry of Land, Infrastructure, Transport and Tourism of Japan (MLIT) [24], the distance traveled for a unit volume of fuel consumed by a vehicle is strongly related to the gross weight of the vehicle. Xiao et al. using statistical data extracted from the mentioned report have concluded that the distance traveled by the vehicle per volume unit of fuel is in inverse proportion to the weight of the vehicle [25]. They utilized this data to formulate a linear regression for calculating fuel consumption rate (FCR) by vehicle based on its weight as follows:

$$Y = 0.0000793X - 0.026 \tag{34}$$

In this equation that is chosen as the basis for calculating the fuel consumption in this study, X is the vehicle weight in kg and Y is the fuel consumption rate in liter/km. In order to put the regression function into practice we adjusted it to our proposed model via the following equation:

$$\rho_{ij}^{l} = 7.93 \times 10^{-5}.[W_l + (y_{ij}^{l} + r_{ij}^{l}).w] - 0.026 \qquad \forall (i,j) \in \boldsymbol{\mathcal{A}}, \forall l \in P \tag{35}$$

where ρ_{ij}^{l} corresponds to Y and is the fuel consumption rate of vehicle type l per unit distance traveled on arc (i,j) and X is comprised of two parts: the vehicle's weight (W_l) plus the cargo's weight ($(y_{ij}^{l} + r_{ij}^{l}).w$) on arc (i,j) which consists of both new and returned goods.

3. Validation of the model

In order to understand the proposed model better and prove its validity, in this section the results obtained by solving an example with evident solution are presented and investigated in a three-stage procedure. To code and solve the example, the solver LINGO 9.0 is used. The proposed model in this paper is formulated to make routing and scheduling decisions of the transportation fleet in a green framework simultaneously. This means that, although the model's constraints are connected, they can be divided into three separate parts: constraints associated with (1) routing, (2) scheduling and (3) environmental costs. In order to validate the proposed model and verify the accuracy of its performance, the model is relaxed to independent sub-models. In each step the preciseness of one of them is shown separately. The expressions of the objective function, constraints and relations used in each sub-problem are classified in Table 3.

Table 3. The characteristics of sub-problems used for validation

Sub-problem / Problem	Terms of objective function	Constraints	Relations
3-1	2, 3, 4	2-11, 29-30	33
3-2	1, 2, 3, 4	2-11, 26-30	31, 32, 33
3-3	1, 2, 3, 4, 5, 6	2-30	31, 32, 33

Note that sub-problem 3-1 is the routing problem including returned goods management, 3-2 is the same plus minimizing the cost of CO_2 emission and problem 3-3 is the model in its complete form. The data used to validate the proposed model are listed in Table 4.

The results of problem 3-1 are shown in table 5. Regarding the capacity of available vehicles, it is obvious that there is only a unique way to meet all demands and that is to assign customer numbers 1 and 3 to vehicles type 1 and customers numbers 2 and 4 to vehicle type 2. Therefore routes 0-1-0 and 0-3-0 are certainly created. Among the possible routes 0-2-4-0 and 0-4-2-0, one should be selected. Since the objective function tries to maximize the revenue gained from collecting returned goods at the same time as it meets all demands, choosing route 0-4-2-0 is the only optimum solution as it can be seen from the solution.

Table 4. The data used to validate the proposed model

i=0,…,4 (i=0 is depot)		φ_i=0,100,200,300,200	
l=1,2	k=1,2	δ_i=0,50,100,150,100	
NV_l=2,1	LD=5	ξ_{1i}=0,0.001, 0.001, 0.001, 0.001	
W_l=4000,8000		ξ_{2i}=0,0.001, 0.001, 0.001, 0.001	
C_l=400,800		b_k=0,2	e_k=2,4
F_l=1000,2000		$L_k=(e_k-b_k)$=2,2	
Q_i=0,350,300,400,460		DT_i=0,0.75,0.75,3,1.5	
R_i=0,30,520,420.40		$D_{ij}=\begin{bmatrix} 0 & 10 & 10 & 20 & 15 \\ 10 & 0 & 15 & 15 & 30 \\ 10 & 15 & 0 & 15 & 15 \\ 20 & 15 & 15 & 0 & 30 \\ 15 & 30 & 15 & 30 & 0 \end{bmatrix}$	
α=2.5,4	c=10		
β=100	w=10		
ω=0.3	M=10^8		
$v_{ij}^k=\begin{bmatrix} 0.001 & 0.001 & 60 & 80 & 60 & 80 & 60 & 50 & 60 & 50 \\ 60 & 80 & 0.001 & 0.001 & 60 & 80 & 60 & 50 & 60 & 80 \\ 60 & 80 & 60 & 50 & 0.001 & 0.001 & 60 & 50 & 60 & 80 \\ 60 & 80 & 60 & 50 & 60 & 50 & 0.001 & 0.001 & 60 & 80 \\ 60 & 80 & 60 & 50 & 60 & 80 & 60 & 80 & 0.001 & 0.001 \end{bmatrix}$			

Table 5. Numerical representation of the solution of sub-problem 3-1

Route 0-1-0	$X_{01}^1=1$	$y_{01}^1=350$	$r_{01}^1=0$
	$X_{10}^1=1$	$y_{10}^1=0$	$r_{10}^1=30$
Route 0-3-0	$X_{03}^1=1$	$y_{03}^1=400$	$r_{03}^1=0$
	$X_{30}^1=1$	$y_{30}^1=0$	$r_{30}^1=400$
Route 0-4-2-0	$X_{04}^2=1$	$y_{04}^2=760$	$r_{04}^2=0$
	$X_{42}^2=1$	$y_{42}^2=300$	$r_{42}^2=40$
	$X_{20}^2=1$	$y_{20}^2=0$	$r_{20}^2=560$

There are two variables y_i and ρ_{ij}^l respectively in the second and third terms of the objective function of sub-problem 3-1 where the first one tries to maximize the number of collected returned goods and the second one tries to do the opposite. Thus, it is expected that a significant decrease of β leads to a decrease in the amounts of y_is. Therefore, in several iterations, every time, the amount of β is reduced to observe the results of this variation on the optimum amounts of returned goods. The results clearly prove the accurate performance of the both sentences of the objective function interacting with each other. Also, as the results in Table 5 indicate, constraints 2-11 are all doing properly and formulated precisely.

Trying to evaluate sub-problem 3-2, other parts to sub-problem 3-1 are added (see Table 3). Regarding the added term to the objective function, it is expected that by keeping β constant and increasing the parameter ω, the amount of y_is is decreased. By executing this test, as it is predictable, with significant increase in costs caused by emissions of carbon dioxide which are load-dependent and keep the revenue from collecting returned goods, the tendency to returning these goods is reduced.

Finally, by adding the remaining terms of the objective function and constraints, the accuracy of the proposed mathematical model in its complete form (problem 3-3) is investigated. The results from solving the complete model are shown in Table 6 and 7.

Table 6. The solutions of routing and scheduling problem – part 1

$\boldsymbol{x(i,j,l)}$	$\boldsymbol{x(i,j,l,k)}$	$\boldsymbol{d(i,j,l,k)}$
x(0,1,1) = 1	x(0,1,1,2) = 1	d(0,1,1,2) = 10
x(0,3,1) = 1	x(0,3,1,2) = 1	d(0,3,1,2) = 20
x(0,4,2) = 1	x(0,4,2,1) = 1	d(0,4,2,1) = 15
x(1,0,1) = 1	x(1,0,1,2) = 1	d(1,0,1,2) = 10
x(2,0,2) = 1	x(2,0,2,1) = 1	d(2,0,2,1) = 10
x(3,0,1) = 1	x(3,0,1,2) = 1	d(3,0,1,2) = 20
x(4,2,2) = 1	x(4,2,2,1) = 1	d(4,2,2,1) = 15
$\boldsymbol{t(i,j,l,k)}$	$\boldsymbol{y(i,j,l)}$	$\boldsymbol{r(i,j,l)}$
t(0,1,1,2) = 0.125	y(0,1,1) = 350	r(0,1,1) = 0
t(0,3,1,2) = 0.4	y(0,3,1) = 400	r(0,3,1) = 0
t(0,4,2,1) = 0.25	y(0,4,2) = 760	r(0,4,2) = 0
t(1,0,1,2) = 0.125	y(1,0,1) = 0	r(1,0,1) = 0
t(2,0,2,1) = 0.1667	y(2,0,2) = 0	r(2,0,2) = 560
t(3,0,1,2) = 0.25	y(3,0,1) = 0	r(3,0,1) = 400
t(4,2,2,1) = 0.25	y(4,2,2) = 300	r(4,2,2) = 40

Table 7. The solutions of routing and scheduling problem – part 2

$\boldsymbol{i}$	$\boldsymbol{a_i}$	$\boldsymbol{l_i}$	$\boldsymbol{E_i}$	$\boldsymbol{T_i}$
0	4	0	0	0
1	2.125	3.875	0	1.375
2	1	1.8333	0	0.25
3	2.95	3.75	0.05	0
4	0.25	0.75	1.25	0

As it is expected the results of this problem are no different from sub-problem 3-1 in section routing. By adding the index k to the main problem and checking the obtained solutions, the precise performance of time-periods dependent constraints which are added to the problem in this step are observable. Eventually, to evaluate the fifth and sixth terms of the objective function, by keeping all the coefficients used in the objective function constant, one time φ_is and another time δ_is are considerably increased. As it can be anticipated, in the first turn the term $\sum_{i=1}^{n} \varphi_i \; . T_i$ and in the second turn the term $\sum_{i=1}^{n} \delta_i \; . E_i$ equals to zero. Also by increasing φ_2 versus φ_4 and considering the amounts $DT_2 = 0.75$ and $DT_4 = 1.5$, the only possible change in the routing section occurs and route 0-4-2-0 as expected is replaced by route 0-2-4-0 as the new optimum route. The comparison between the three provided examples is an instance of comparing the results of several small-sized problems solved of this kind which shows that the proposed mathematical model is appropriately formulated.

4. META-HEURISTIC SOLUTION APPROACHES

Many combinatorial optimization problems including VRP which are classified as NP-hard cannot be solved for real-world large-sized problems in a reasonable computational time. Therefore, to solve such problems, heuristics and meta-heuristic approaches are needed in order to achieve optimum or near-optimal solutions in a reasonable computational time. Hence, in this paper a simulated annealing (SA) and a genetic algorithm (GA) are presented to deal with this issue.

4.1. Simulated annealing

Simulated annealing was first introduced in 1983 by Kirkpatrick et al [26]. It is one of the first meta-heuristic approaches that is not based on natural evolutionary unlike the usual methods until then. It is a technique for solving unconstrained and bound-constrained optimization problems. The method models the physical process of melting a solid material and then slowly decreases the temperature to reduce defects and minimizes the system energy. The fundamental idea of this technique is a random search procedure using Markov chain. In each iteration of the SA algorithm, a new solution which is called "the neighbor solution" is randomly generated.

4.2. Genetic algorithm

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. It was initially discussed by Holland in 1975 as one of the first introduced meta-heuristic methods [27]. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parent chromosomes to produce the offspring for the next generation. Genes are transferring from current chromosomes to the ones from next generation using mutation and crossover operators. Over successive generations, the population evolves toward an optimal solution.

4.3. The proposed meta-heuristics for the represented problem

To solve the described green vehicle routing and scheduling problem, a simulated annealing and a genetic algorithm are put into practice to deal with the large-sized real-world problems.

4.3.1 Mapping the solution representation

In this section the applied approach to display a random solution including a routing and its corresponding scheduling is described in detail. Solution mapping is one of the most important steps for implementing a meta-heuristic. The encoded solution of the proposed problem is represented in the form of a matrix called chromosome which contains a number of sub-matrices. Each individual chromosome is made up of unique arrays that make it different from the other ones. A random chromosome as depicted in Fig. 5, where the sub-matrices are marked with different colors, is explained as the following: The described chromosome is a randomly generated one using data from Table 4. The first sub-matrix from the chromosome is called the matrix $R = [0, R_l^1|\{0\}, 0, R_l^2|\{0\}, 0 \ldots 0, R_l^h|\{0\}, 0]$ which represents the routing section of the solution (the one on the top-left). Each matrix $R_l^c = [rt^c(1),\ rt^c(2), \ldots, rt^c(\psi_c)]$ indicates a distinct route in which $c = 1,2, \ldots, h$ is the representative of route number and $l = 1,2, \ldots, p$ points out the type of vehicle used to travel the route. Also, $rt^c(j)$ specifies the number of jth visited node on track and

ψ_c denotes the number of nodes on the cth path. The depot as is always determined with zero is the first and the last visited node of every route and the other in-between nodes represent the customers. The second sub-matrix from the represented chromosome is named the matrix $S = [S^1, S^2, \dots, S^h]$ which describes the scheduling section of the solution (the one on the middle-left). Each binary matrix

$S^c = \left[s^{c,1}_{1,2}, s^{c,1}_{2,3}, \dots, s^{c,1}_{j,j+1}, \dots, s^{c,1}_{\psi_c-1,\psi_c} \;;\; s^{c,2}_{1,2}, s^{c,2}_{2,3}, \dots, s^{c,2}_{j,j+1}, \dots, s^{c,2}_{\psi_c-1,\psi_c} \;;\; \dots \;;\; s^{c,k}_{1,2}, s^{c,k}_{2,3}, \dots, s^{c,k}_{j,j+1}, \dots, s^{c,k}_{\psi_c-1,\psi_c}\right]$ depicts the travel scheduling of the cth route where $k = 1,2, \dots, m$ determines the number of available time periods. Also, j and $j + 1$ show the order of visiting for two consecutive nodes such that for each route, $j = 1$ and $j = \psi_c$ (the first and the last visited nodes of each route) are correspond to the depot. Each array $s^{c,k}_{j,j+1}$ takes the value 1 if the connecting arc between jth and $j + 1$th nodes on the route c is traveled in time period k and otherwise it takes the value 0, $(j = 1,2, \dots, \psi_c - 1 \;,\; \forall c, \forall k)$. In a matrix S, each column represents the arc between jth and $j + 1$th visited nodes on the route and each row denotes a time period k. The third sub-matrix shows the allocation of vehicles to the routes (the one on the top-middle). The fourth one demonstrates the amount of returned goods collected from each customer (the one on the top-right) and the fifth one is the sub-matrix of arrival and departure times to/from the nodes (the one on the bottom-left). In the recent sub-matrix, each column contains the scheduling of entry and exit associates with an arc (i, j) corresponding to the columns of sub-matrix S. Thus, in each column, each element of the first row shows the departure time from node i, each element of the second row represents the arrival time to the node j and every element of the third row denotes the departure time from node j.

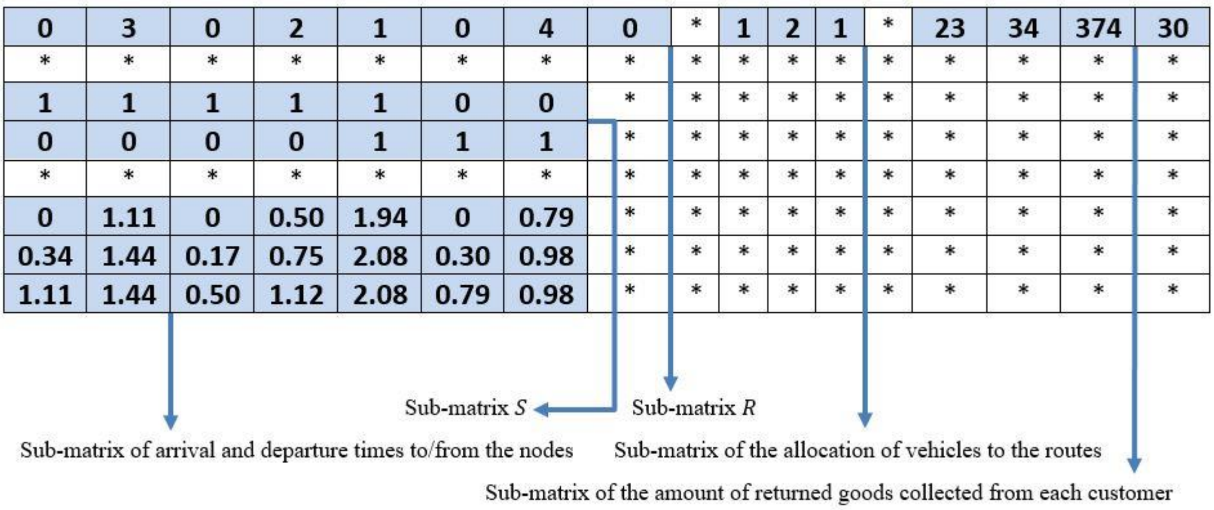

Fig. 5 A randomly generated chromosome

To clarify the explanations, refer back to figure 5, in order to meet all demands, three separate routes are considered. For example, in the routing section, it can be seen that the second route ($c = 2$) is traveled by the vehicle type two ($l = 2$). This path is as follows: $R^2_2 = \{rt^2(1) = \mathbf{0}\,, rt^2(2) = \mathbf{2}\,, rt^2(3) = \mathbf{1}\,, rt^2(4) = \mathbf{0}\}$. The vehicle starts to travel its route from the depot (node 0) and after visiting nodes 2 and 1 returns to the depot at the end of its path. The content of matrix S concerning the same route states that the arc between the depot and node number 2 and the arc between nodes 2 and 1, both are traveled in time period 1 and the arc connecting node 1 to the depot is traveled in the first and second time periods together (columns 3rd to 5th from the matrix S). Also, the number of returned goods that is determined to be collected from nodes 2 and 1 are 34 and 374 respectively.

Finally, the arrival and departure times to/from each node on this path can be observed in the three bottom rows of the proposed chromosome, columns 3 to 5.

For each specific problem, the number of arrays in routing section is constant and showed by (n_t). It is calculated via the relations as described in the following: Each route always contains two fixed zeros at its beginning and end which are referred to as $n_{fixed} = 2$. The maximum number of possible routes for a particular problem is calculated through the equation (36). This amount equals to the minimum number of customers (n) and the total number of all available vehicles in the fleet.

$$max(R) = min\ \left(n\ , \sum_{l=1}^{p} NV_l\right) \tag{36}$$

Equation (37) determines the number of in-between zeros ($n_{mid.zeros}$) that is also constant for a given problem. These zeros which are located between the first and last fixed zeros play the role of separators between distinct routes in each solution.

$$n_{mid.zeros} = max(R) - 1 \tag{37}$$

Eventually, the number of arrays represents the routing section (or matrix R) of each solution in a specific problem is calculated via:

$$n_t = n + n_{fixed} + n_{mid.zeros} \tag{38}$$

By considering that in each unique matrix R, each in-between zero that separates two different routes is in common between both of them, if the number of $n_{mid.zeros}$ arrays is subtracted from the total number of arrays of all routes, the rest of difference is also equal to n_t:

$$n_t = \sum_{c=1}^{max(R)} \sum_{j=1}^{\psi_c} j\ -\ n_{mid.zeros} \tag{39}$$

As a result, the dimensions of the matrix R is: $1 \times n_t$. Also, calculation associated with the size of matrix S is as follows:

$$Columns_S = \sum_{c=1}^{max(R)} \left(\sum_{j=1}^{\psi_c} j - 1\right) \tag{40}$$

The above equation calculates the number of required columns of the matrix S in each individual solution. Therefore, the dimensions of the matrix S is: $m \times Columns_S$. It can be proved that relation (41) is always established between n_t and $Columns_S$.

$$n_t = Columns_S + 1 \tag{41}$$

It should be mentioned that the dimensions of matrices R and S for each particular problem which are calculated through the stated equations above is always constant. It should be considered that if two or more in-between zeros are placed next to each other, since the arc between two zero nodes is virtual and in practice it is not able to be formed, the columns in matrix S that correspond to the scheduling of these type of virtual arcs are always composed of zero arrays and are not included in calculations of scheduling. To have feasible matrices R and S the following constraints (42-46) must be satisfied:

$$R' - \{0\} = \mathcal{N} - \{0\} \tag{42}$$

$$rt^c(1) = 0\,, rt^c(\psi_c) = 0 \quad , \quad \forall c \tag{43}$$

$$\sum_{k=1}^{m} s_{j,j+1}^{c,k} \geq 1 \quad , \quad \forall c\,, j = 1,2,\dots,\psi_c - 1 \tag{44}$$

$$1 - s_{j,j+1}^{c,k} \geq s_{j',j'+1}^{c,k'} \qquad \forall c\,, j = 1,2,\dots,\psi_c - 1\,, j' > j\,, k' < k \tag{45}$$

$$1 - s_{j,j+1}^{c,k} \geq s_{j',j'+1}^{c,k'} \qquad \forall c\,, j = 1,2,\dots,\psi_c - 1\,, j' < j\,, k' > k \tag{46}$$

Let R' the set of numbers in sub-matrix R. Constraint (42) shows that all customers should be visited only once. Constraint (43) states that the first and last node of each individual route is the depot. Constraint (44) guarantees that all arcs of each route are traveled at least in one time period. Constraint (45) ensures if $s_{j,j+1}^{c,k} = 1$, all elements on its top-right corner in the matrix S have to be zero; likewise, constraint (46) does the same if $s_{j,j+1}^{c,k} = 1$ for its bottom-left corner. Below is a rule we employ while implementing meta-heuristics:

$$\sum_{k=1}^{m} s_{j,j+1}^{c,k} = A_{j,j+1}^{c} \quad , \quad \forall c\,, j = 1,2,\dots,\psi_c - 1 \tag{47}$$

$$\sum_{\substack{k=\tau+A_{j,j+1}^{c} \\ if\ A_{j,j+1}^{c}>1}}^{m} s_{j,j+1}^{c,k} = 0 \quad , \quad \forall c\,, j = 1,2,\dots,\psi_c - 1 \tag{48}$$

Equation (47) calculates the cumulative value obtained for each column of each route in the matrix S and names it $A_{j,j+1}^{c}$. Also, for each route c, let τ the first time period in which the arc $(j, j+1)$ is traveled. Equation (48) is established in case of any arc is traveled in more than a single time period. It points out that those periods are located immediately after each other. In other words, in individual columns of the matrix S, wherever number 1 is repeated more than once, they come right after each other in consecutive rows. This rule is employed because it reduces the search space and based on our experiment, for the presented test problems leads to better results in the same computational time compared to the case it is not used, e.g., for a test problem with $n = 9$, $m = 3$, $p = 3$ and number of 10 runs, applying this rule resulted in a 6.2% and 3.5% reduction in the objective value for SA and GA respectively. In each feasible solution, for each single path in matrix R, there is a unique corresponding path in matrix S which starts from the top-left corner and continues its way to the bottom-right corner. This path does not necessarily start at the first period and end at the last one. The path must be continuous through the columns, whereas it can be non-continuous through the rows except for the rows of each unique column. Finally, it should be mentioned that the maximum number of arcs that might be traveled in more than a single time period is equal to $m - 1$ in each path and $h.(m - 1)$ for each solution.

4.3.2 Generating a random solution

The initial step for implementing either SA or GA is to establish a function that generates random solutions. To do so, the minimum number of necessary routes is calculated to form a feasible solution based on the number of available vehicles of each capacity and the quantity of demands. Then the maximum number of possible routes is calculated which is equal to the minimum of the number of customers and the number of available vehicles in fleet. The number of routes to generate a solution is randomly selected from the interval lying between the minimum number of necessary routes and the maximum number of possible routes. At this point, a random matrix R and its corresponding vehicle assignments to routes are produced. The next step is to determine the amount of returned goods collected from each customer. These amounts are specified according to the vacant capacity of vehicles on the roads and the amount of potential returned products on that node (R_i). The minimum of these two values is equal to the maximum number of returned goods which the vehicle can actually retrieve from that node. If the obtained minimum is greater than zero, a random number between zero and the achieved minimum value determines the exact amount of returned goods collected from that node; otherwise the amount is equal to zero. This procedure guarantees that the amount of retrieving returned goods from each customer never exceeds the capacity of its visiting vehicle. In the final step, a random matrix S corresponding to the existing routes is generated through a pre-designed random procedure and the calculations related to the arrival and departure times to/from each node are carried out subsequently.

4.3.3 Fitness evaluation

At this stage, a function is defined to calculate the objective value named the cost function or so called the fitness function. It is utilized to evaluate the fitness of every single chromosome generated in each individual iteration of the proposed meta-heuristics and compare them to one another in terms of quality and optimality. The various costs that are formulated in the fitness function in the form of mathematical expressions are as follows: the cost of total CO_2 emissions, fixed costs, variable costs including fuel consumption cost and total costs of tardiness and earliness minus the total revenue gained by collecting returned goods. Two additional costs are added to the fitness function in the form of penalizing strategy. The first one is the penalty for infeasible solutions that is imposed to the fitness function if a random generated solution is infeasible. Therefore, the total weight of goods assigned to each vehicle in a random generated solution is calculated and a penalty equal to M is considered per violation of the capacity of a vehicle. It is called $Penalty\ 1$ and calculated as follow:

$$Penalty\ 1 = M \times \Pi \tag{49}$$

where Π represents the number of times when a violation of the capacity of a vehicle is occurred. The second cost is the penalty for each exceeding of the time available for each period. Whenever in a solution, the total spent time to travel different arcs in a single time period on any route exceeds the duration of that time period, a penalty called $Penalty\ 2$ is added to the fitness function as follows:

$$Penalty\ 2 = \Lambda . \sum_{c=1}^{h} \sum_{k=1}^{m} max \left\{ 0\, , \sum_{j=1}^{\psi_c - 1} \mathrm{T}_{j,j+1}^{c,k} - (e_k - b_k) \right\} \tag{50}$$

Where $\mathrm{T}_{j,j+1}^{c,k}$ is a time portion in which arc $(j, j+1)$ from path c is traveled within time period k . Also, Λ is the penalty coefficient for any overuse of a time period. The sum of all mentioned costs from the objective function of the proposed mathematical model plus the two penalties 1 and 2, form the fitness function.

4.3.4 Neighbor solutions and offspring

SA is a point-based algorithm while GA is a population-based one. Hence, there are several known operators available to generate new solution(s) which is a neighbor solution for the SA and are offspring for the GA, in order to make a comparison between the current solution(s) and the new one(s), using fitness function. Due to the nature of the present problem, permutation mutation and crossover operators are applied in this paper. Four of the most commonly used mutation operators: swap, inversion, reversion and scramble along with a fifth one which is a particular form of the swap operator and is modified specially for this problem are used by the SA to produce neighbor solutions in each iteration. Only one of the mentioned operators is randomly selected and applied to the matrix R. The modified swap operator is different from the other ones in the way that by executing each of the four ordinary operators on sub-matrix R, there is a probability of a change in the number of paths, whereas the modified swap operator is changed so that after applying it to any given solution, only two customers are displaced such that the number of paths of the neighbor solution and the number of customers located on each path do not differ from the current solution and therefore unlike the other operators, there is no need for new assignments and calculations which results in less computational time. In order to generate offspring for GA, in addition to the described mutation operators, four different crossover operators are also designed exclusively. Single-point, double-point and four-point crossovers which are named based on the number of crossover points and uniform crossover that has an undetermined number of cutting points. In addition to the parents, the last operator uses a third chromosome which is called “mask chromosome”. It is used to randomly decide which genes from each parent should be transferred to the offspring. In the same way as SA, GA also uses only one operator per iteration to generate offspring.

There is a modified procedure considered for the times when duplicate genes are observed in the next generation chromosomes. This situation may only occur when a crossover operator is applied. After using any of the operators, if the number of routes is changed in the neighbor solution or the offspring, first, vehicle allocation to the new paths are done; then, the calculations of the amounts of returned goods and also scheduling are performed with respect to new routes. Otherwise, there is no need to reallocate vehicles to new routes and it can be carried out with a predetermined probability. However, the calculations of returned goods and new scheduling must certainly be done considering relocations of genes in new solution(s) compared to the current one(s).

5. Experimental design and parameters setting

In this section, some numerical experiments are set up to evaluate the proposed algorithms, their effectiveness and efficiency and compare their performance to each other in terms of solutions quality and computational times. For this, a set of test problems is generated, then the parameters of both algorithms are set using Taguchi method. T-Tests are then presented and finally, sensitivity analysis is carried out.

5.1. Data Generation

In order to evaluate the performance of the proposed algorithms, experimental calculations are done to compare the effectiveness of the solutions produced by proposed meta-heuristics, with each other and with the best solutions obtained by the exact solver LINGO 9.0. In these comparisons, two factors: solution quality (best objective function value) and computational times are considered. To compare different methods, a set of test problem is applied. To demonstrate the effectiveness of the proposed algorithms, various-sized problems are considered. The generated test problems are classified into three categories based on their size and listed in Table 8. $N15$ can be mentioned as a good example of real-world applications where four types of vehicles considering their gross weights is very common and 4 two-hour time periods simulates a regular 8 hour working day.

Table 8. The generated test problems

n	*p=2*			*p=3*			*p=4*		
	m=2	*m=3*	*m=4*	*m=2*	*m=3*	*m=4*	*m=2*	*m=3*	*m=4*
4	*N01*	*N02*		*N03*					
5	*N04*				*N05*				
6									
7									
8	*N06*								
9		*N07*			*N08*				*N09*
10									*N11*
11									
12	*N10*								
13									
14		*N12*							
15									
16				*N13*					
17									
18					*N14*				
19									
20									*N15*

Small-sized	Medium-sized	Large-sized

Intervals and distribution functions are generated as follows: to generate the elements of matrix v_{ij}^{k} , the discrete uniform distribution [2,8] is used and generated values are multiplied by 10. The elements of D_{ij} are obtained by multiplying 5 times the numbers generated by the discrete uniform distribution [2,6]. Q_is and R_is are distributed in the ranges of [100,500] and [10,520] respectively. DT_is are selected through the values $\{1, 1.5, 2, 2.5, 3, 3.5, 4\}$. Also, the elements of matrices δ_i and φ_i are taken the values $\{40, 50, 100, 150\}$ and $\{100, 150, 200, 300\}$ respectively.

5.2. Parameters setting

Parameters setting is one of the most important procedures for designing meta-heuristics and has a significant effect on achieving optimal solutions. Since different values of controller parameters of these algorithms affect two major computational indices, the solution quality and the solution time, generally a set of experiments named calibration is carried out in order to determine the optimal combination of these controller parameters. The main SA's controller parameters are: number of algorithm iterations (G_{max}), initial temperature (T_0) and cooling rate ($Alpha$). Also the same ones for GA are population size (Pop_{size}), maximum number of generations (G_{max}), crossover probability (P_c) and mutation probability (P_m). Usually each named parameters affects computational indices within a specific interval of its values. Computational indices used in this section are the mean of optimal solutions of the objective function and the mean of their computational times in 50 separate iterations per parameter of each algorithm. Five distinct test problems with different combinations of n, p and m are applied for this purpose. In this section, first, the effective range for each parameter is specified relatively, then by designing multifactorial experiments, the interactions of those parameters is analyzed and eventually the optimal combination is estimated.

At this point, a number of test problems is arranged to find the effective range for each controller parameter individually. Each parameter of both algorithms is tested separately and the outcomes are as follows: For SA parameters, the results for $Alpha$ show that solutions are at their best quality level in the interval [0.94, 0.99]. The effective range for G_{max} is [2000, 3000] and for T_0 the best solution quality is set in the interval [50, 100]. For GA parameters, G_{max} performs its best while taking values from the interval [100, 200]. Proper amounts for Pop_{size} are 30, 50 and 80. Also, results show that the suitable ranges for P_c and P_m are [0.85, 0.95] and [0.04, 0.08] respectively.

In the last stage, to determine the best values of each parameter, a design of experiments (DOE) based on Taguchi's method using data obtained from the previous stage is set. Since classical DOE procedures are complicated to execute and also the number of required experiments is increased exponentially as the number of parameters grows, applying Taguchi's technique is very helpful to avoid these challenges [28]. It can simply be carried out by running a limited number of experiments using a special design of orthogonal arrays to investigate the entire space of parameters quickly and find the best values in much shorter computational time. Eighteen different tests for SA and twenty-seven different tests for GA are performed and each one is repeated ten times. For each algorithm the total normalized form of the mean values obtained for the objective function and the computational times is utilized as the decision criteria in the final analysis to find the optimum combination of parameters.

The signal-to-noise ratio (S/N) in Taguchi's method shows the variation of the response variable. The aim is to find the maximum value of (S/N) ratio because it minimizes the effect of uncontrollable or so-called noise factors and makes the solution more robust. In this study, "the smaller the better" rule is applied which is calculated via the following relation:

$$S/N\ ratio = -10 \times \log_{10}\left(\frac{1}{n}\sum_{i=1}^{n} {y_i}^2\right) \tag{51}$$

where y_i is the normalized deviation of the values of objective function from the least obtained value for each test condition (i) [29]. Taguchi's DOE for each algorithm is performed separately using MINITAB 16 and the results are displayed in Table 9. It should be noted that the population size has the most influence on solutions quality and computational times for either SA or GA; The initial temperature for SA and the mutation probability for GA have the least effects on the same outcome indices as well.

Table 9. The best values of the controller parameters of the proposed algorithms

Algorithm	Factor	Best value
SA	$Alpha$	0.98
	G_{max}	2000
	T_0	50
GA	G_{max}	100
	Pop_{size}	30
	P_c	0.9
	P_m	0.06

5.3. Experimental results

The proposed mathematical model is coded and run using the exact solver LINGO 9.0 via its global solver. The proposed algorithms are coded using MATLAB R2014a and run on a computer Core 2 Dou, 2.53 GHz CPU with 4 GB RAM. Each problem from Table 8 is run 30 times and the best solution is selected. This process is performed per algorithm separately. Therefore, a total of $30 \times 15 \times 2 = 900$ runs are carried out. Also, the maximum computational time for solving the model using LINGO is considered one hour for small and medium-sized instances and five hours for large-sized ones. If the optimal solution is not reached within this period, the best solution obtained until then (if any achieved) is considered. These restrictions are imposed to computational times since the model is so complicated and even by passing much more time, LINGO is not able to find the optimum solution, e.g. for problem $N07$ the best objective value achieved after three hours is still the same as the one obtained after the first hour or for problem $N12$, it can be seen that after 5 hours from the beginning of the search, LINGO is still not able to find any feasible solution. The numerical comparison between the performances of LINGO versus the proposed algorithms is shown in Table 10. Eventually, it should be mentioned that the classification of problems in Table 8 into three categories based on their size is also done using Taguchi's DOE method. According to the results, it is found that indices k, i and l respectively have the highest impacts on both objective values and computational times.

Table 10. The comparison between the performance of LINGO and the proposed algorithms

Problem Name	**LINGO**		**Simulated Annealing (SA)**					**Genetic Algorithm (GA)**				
	Best solution	Time (sec)	Best solution	%PRE	%RPD	Average	Mean CPU Time	Best solution	%PRE	%RPD	Average	Mean CPU Time
N01	1871.56	55	2149.66	14.86	8.91	2326.40	7.94	2136.00	14.13	10.28	2355.74	4.45
N02	2027.67	56	2223.53	9.66	7.44	2389.15	8.63	2259.49	11.43	9.84	2442.49	4.71
N03	2171.56	361	2178.35	0.31	11.63	2431.68	8.30	2214.89	2.00	12.46	2449.91	4.58
N04	2842.22	354	2842.23	0.00	6.43	3024.96	8.18	2852.52	0.36	7.90	3066.78	4.71
N05	2989.43	2716	3048.74	1.98	7.31	3239.82	9.74	3018.97	0.99	9.80	3314.90	5.59
Mean	**2380.48**	**708.40**	**2488.50**	**4.54**	**7.79**	**2682.40**	**8.56**	**2496.37**	**4.87**	**9.54**	**2725.96**	**4.81**
Mean %PRE - %RPD				**5.36**	**8.34**				**5.78**	**10.06**		
N06	8543.22	3600	5585.27	-	9.24	6101.42	12.13	5702.66	-	11.40	6222.39	7.28
N07	11368.02	3600	6052.86	-	21.62	7361.51	14.85	6880.45	-	26.10	7632.56	8.96
N08	-	3600	6527.45	-	16.07	7574.96	14.77	6526.35	-	18.79	7752.64	9.10
N09	-	3600	6877.06	-	22.21	8386.95	15.22	6862.36	-	26.62	8689.40	9.89
N10	-	3600	9949.69	-	9.12	10857.61	13.76	10207.97	-	14.52	11395.25	8.81
Mean			**6998.47**	**-**	**15.12**	**8056.49**	**14.15**	**7235.96**	**-**	**19.15**	**8338.45**	**8.81**
Mean %RPD					**15.65**					**19.47**		
N11	-	18000	9505.33	-	12.43	10660.29	17.59	9481.28	-	10.40	10468.21	10.89
N12	-	18000	11108.74	-	12.04	12446.65	17.52	11678.32	-	15.18	12795.72	10.82
N13	-	18000	15024.42	-	10.98	16673.72	18.92	16114.60	-	15.36	17332.98	11.27
N14	-	18000	18790.40	-	7.21	20145.26	21.02	20047.93	-	17.16	22014.93	13.87
N15	-	18000	29013.66	-	6.40	30870.93	23.81	29647.17	-	12.67	32690.23	15.77
Mean			**16688.51**	**-**	**8.81**	**18159.37**	**19.77**	**17393.86**	**-**	**14.21**	**19060.41**	**12.52**
Mean % RPD					**9.81**					**14.15**		

5.4. Algorithms evaluation

In order to evaluate the performance of the proposed algorithms, two criteria are considered. Since the five small-sized problems out of fifteen designated test problems achieve the optimum solutions, the percentage relative error (PRE) criterion is established to evaluate the performance of the proposed algorithms for such-sized examples. PRE is defined as follows:

$$PRE = \frac{|Z_t - Z_m|}{Z_t} \times 100 \tag{52}$$

In the above equation, Z_t is the optimum objective value and Z_m is the best solution value obtained by an algorithm. For medium and large-sized problems, the relative percentage deviation (RPD) criterion is used to evaluate the performance of the proposed algorithms. For each problem, RPD compares the mean objective value obtained by the selected algorithm (Z_{mean}) to the best objective value found among the two algorithms (Z_{min}) as follows:

$$RPD = \frac{Z_{mean} - Z_{min}}{Z_{min}} \times 100 \tag{53}$$

As it can be seen for small-sized problems, SA has 5.36% and GA has 5.78% average difference with the optimum solution. While the quality of solutions obtained from SA and GA are 4.54% and 4.87% lower on average compared to the ones achieved by LINGO, but in terms of computational times SA operates more than 80 times and GA more than 145 times faster on average than LINGO. It is evident that such a reasonable quality loss in exchange for such an impressive reduction in solution time indicates the satisfactory performance of the proposed algorithms in achieving near-optimal solutions in a reasonable time. Also, since penalizing is the main strategy applied to deal with constraints in this paper, it should be noted that the strategy leads to feasible solutions in almost all small-sized problems and more than 90% of medium-sized problems. As the dimensions of the problem increase, for large-sized problems, the rate of achieving feasible solutions obtained in the experimental results varies from 60% to 70% of the instances. Fig. 6 compares the objective values of LINGO, SA and GA and Fig. 7 does the same for computational times.

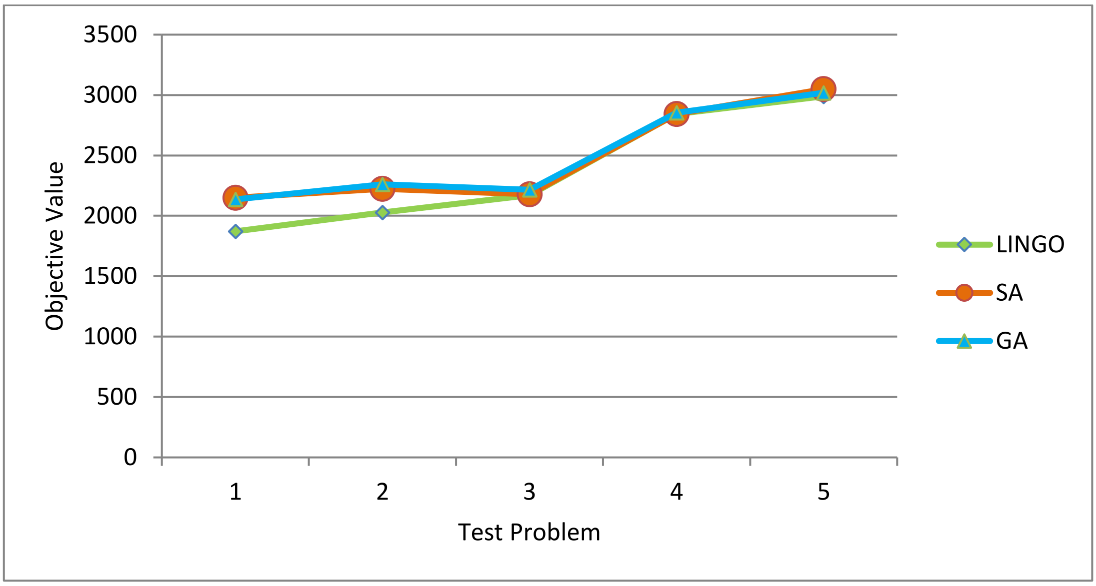

Fig. 6 The comparison between the objective values of LINGO, SA and GA

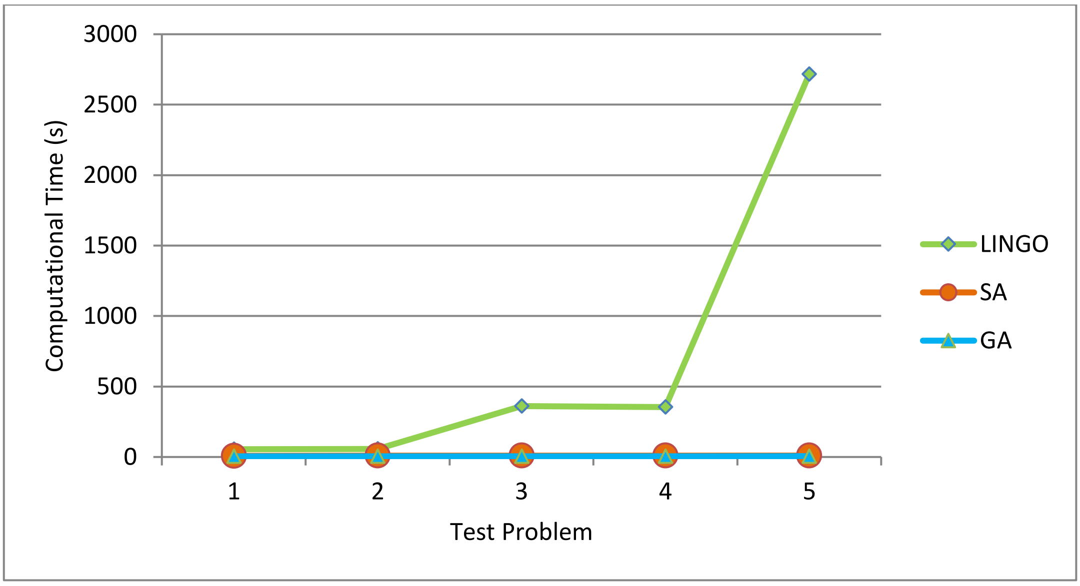

Fig. 7 The comparison between the computational times of LINGO, SA and GA

As for middle-sized problems, the mean difference percentage between the best objective value found among the two algorithms and the mean objective value obtained by SA is 15.65% and by GA is 19.47%. Also, the same criterion for large-sized instances is calculated 9.81% for SA and 14.15% for GA which indicates more relative convergence for both algorithms especially SA as the scale of problems becomes larger. The comparisons of the solutions quality of the proposed algorithms as well as their corresponding CPU times are depicted in Fig. 8-11.

Fig. 8 The comparison of relative percentage deviation; SA vs. GA

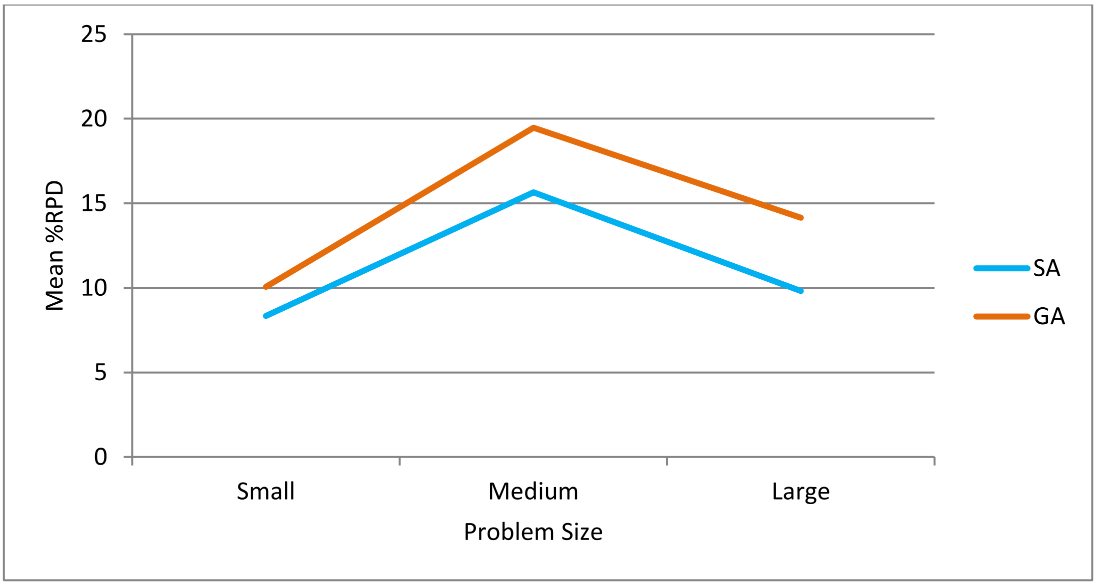

Fig. 9 The comparison of mean relative percentage deviation for problems of all sizes; SA vs. GA

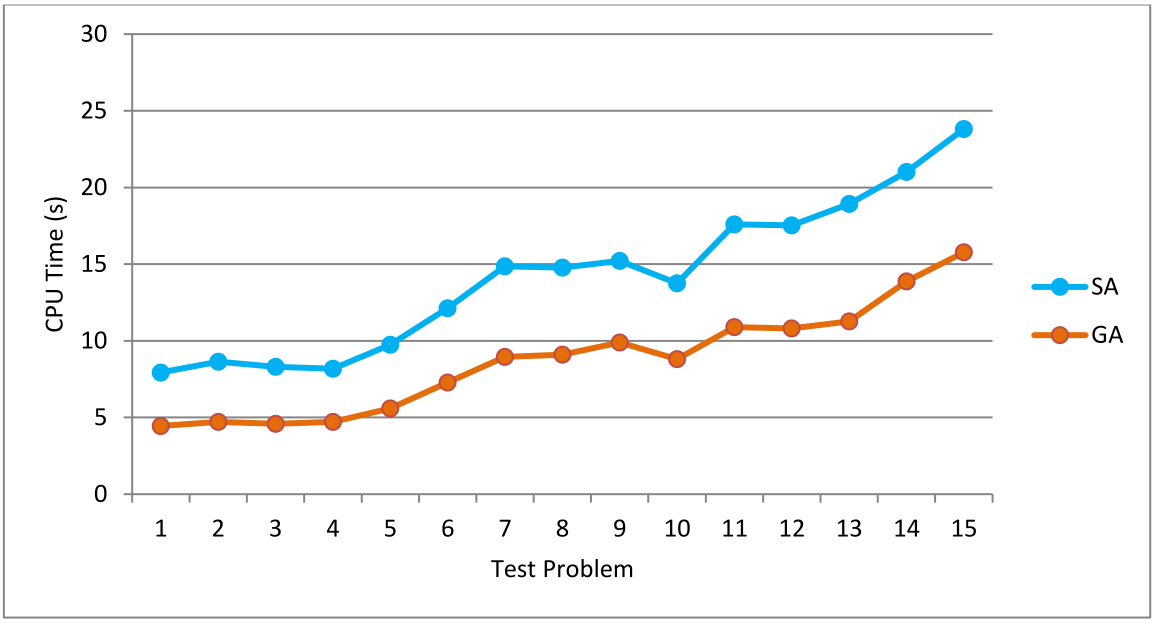

Fig. 10 The comparison of central processing unit (CPU) times; SA vs. GA

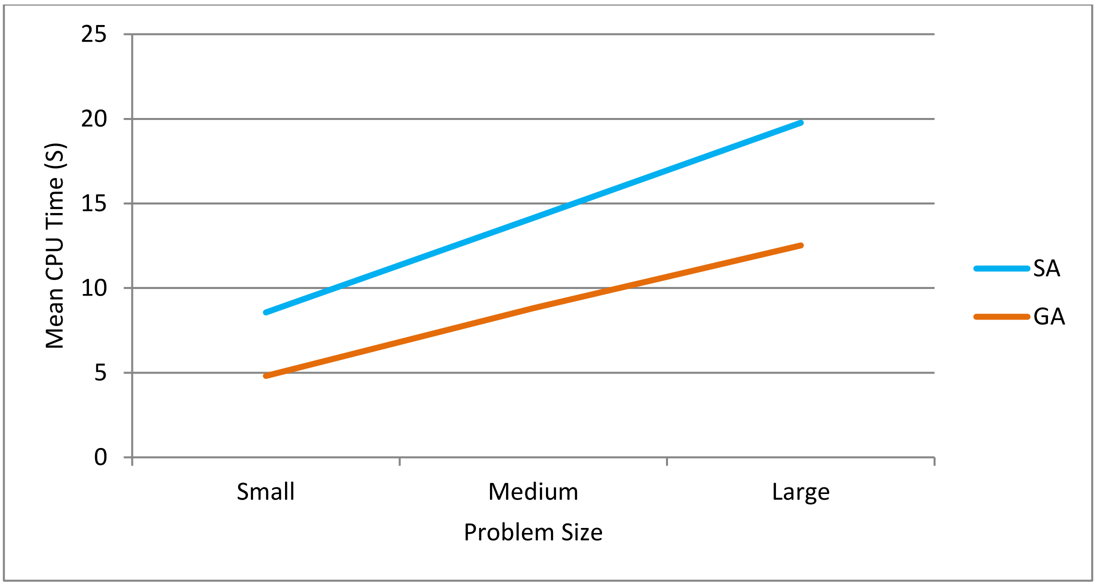

Fig. 11 The comparison of mean central processing unit (CPU) times for problems of all sizes; SA vs. GA.

Fig. 12-13 show examples of convergence curves of the presented algorithms to observe their behaviors more comprehensively. The test problem N08 is applied. Computational times for SA and GA are 13.84 and 9.19 seconds, respectively.

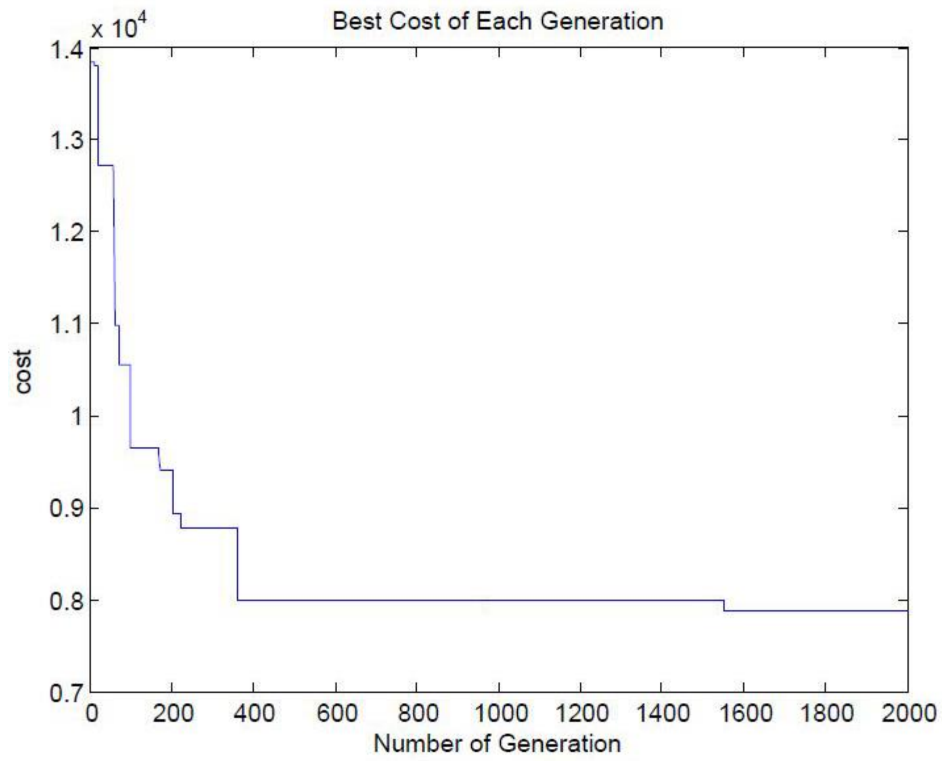

Fig. 12 An example of convergence curve for the test problem *N08*; SA

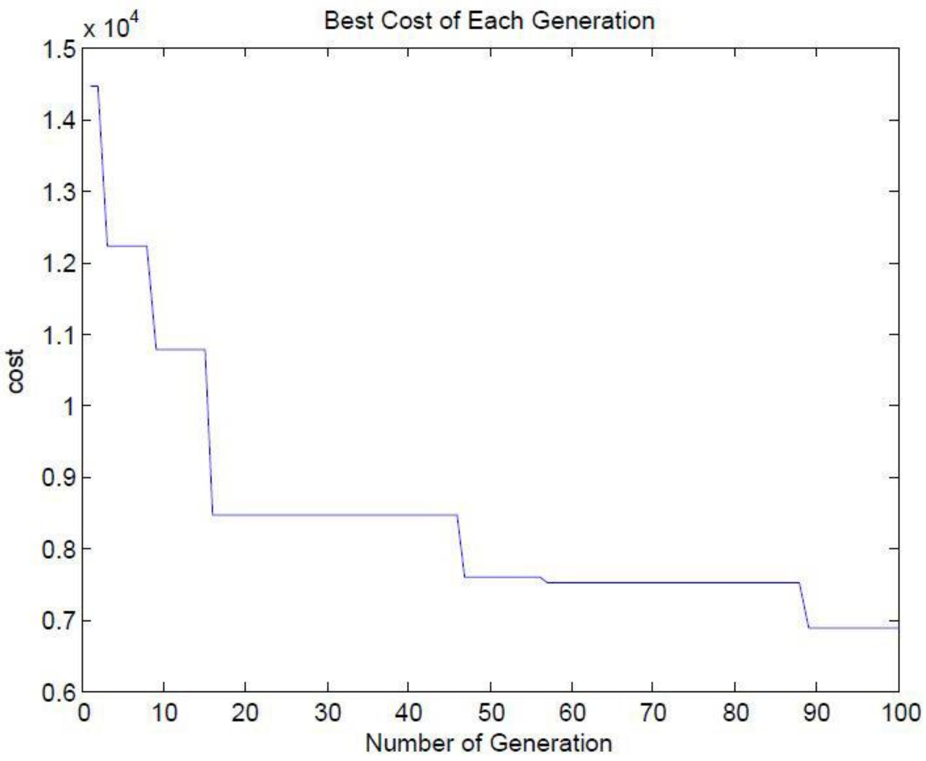

Fig. 13 An example of convergence curve for the test problem *N08*; GA

Two statistical hypothesis testing (T-test) are carried out on the outcomes of the algorithms in terms of objective values and computational times. The applied independent-samples T-test procedure tests the difference between the means of two algorithms in order to find out whether it is significant or not. The procedure also displays descriptive statistics for each test variable and a test of variance equality, namely Levene's test. The test is performed for all test problems using SPSS 16.0 as follows:

$$\begin{cases} H_0: \mu_{SA} = \mu_{GA} \\ H_1: \mu_{SA} \neq \mu_{GA} \end{cases} \tag{54}$$

where, the confidence interval for the difference between the two variables is considered 95%. The results are shown in Tables 11 and 12. Mean difference is obtained by subtracting the sample mean for GA from the sample mean for SA and finally the 95% confidence interval of the difference provides an estimate of the boundaries between which the true mean difference lies in 95% of all possible random samples of 60 iterations per a problem. The results are listed in Tables 11 and 12.

Table 11. The results of T-test for objective values

Group Statistics						**Independent Samples Test**								
						Levene's Test for Equality of Variances		**t-test for Equality of Means**						
													95% Confidence Interval of the Difference	
Problem Name	Algorithm	N	Mean	Std. Deviation	Std. Error Mean	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
N01	SA	30	2.3264E3	106.04	19.36	1.454	0.233	-0.960	58	0.341	-29.34	30.55	-90.50	31.81
	GA	30	2.3557E3	129.46	23.63									
N02	SA	30	2.3892E3	96.53	17.62	1.432	0.236	-2.063	58	0.044	-53.33	25.85	-105.09	-1.85
	GA	30	2.4425E3	103.61	18.91									
N03	SA	30	2.4317E3	99.34	18.13	1.884	0.175	-0.607	58	0.546	-18.22	30.00	-78.29	41.84
	GA	30	2.4499E3	130.94	23.90									
N04	SA	30	3.0250E3	99.53	18.17	1.440	0.235	-1.306	58	0.197	-41.82	32.02	-105.92	22.28
	GA	30	3.0668E3	144.44	26.37									
N05	SA	30	3.2398E3	111.95	20.44	3.352	0.072	-2.134	58	0.037	-75.07	35.18	-145.50	-4.64
	GA	30	3.3149E3	156.87	28.64									
N06	SA	30	6.1014E3	238.89	43.61	2.296	0.135	-1.747	58	0.086	-120.97	69.24	-259.59	17.64
	GA	30	6.2224E3	294.60	53.78									
N07	SA	30	7.3615E3	560.81	102.39	6.966	0.011	-2.249	48.466	0.029	-271.05	120.51	-513.314	-28.79
	GA	30	7.6326E3	348.19	63.57									
N08	SA	30	7.5750E3	435.70	79.54	0.429	0.515	-1.454	58	0.151	-177.67	122.187	-422.26	66.90
	GA	30	7.7526E3	507.98	92.74									
N09	SA	30	8.3870E3	975.46	178.09	2.965	0.090	-1.344	58	0.184	-302.45	224.98	-752.80	147.90
	GA	30	8.6894E3	753.00	137.47									
N10	SA	30	1.0858E4	498.88	91.08	0.058	0.811	-4.459	58	0.000	-537.64	120.58	-779.02	-296.25
	GA	30	1.1395E4	432.84	79.02									
N11	SA	30	1.0660E4	917.96	167.59	0.424	0.517	0.828	58	0.411	192.08	231.84	-272.00	654.17
	GA	30	1.0468E4	877.45	160.20									
N12	SA	30	1.2447E4	462.66	84.47	1.697	0.198	-2.644	58	0.011	-349.06	132.03	-613.36	-84.76
	GA	30	1.2796E4	555.83	100.48									
N13	SA	30	1.6674E4	671.17	122.53	0.222	0.639	-3.732	58	0.000	-659.26	176.66	-1012.88	-305.63
	GA	30	1.7333E4	696.98	127.25									
N14	SA	30	2.0145E4	784.75	143.27	0.000	0.994	-9.089	58	0.000	-1869.66	205.70	-2281.43	-1457.89
	GA	30	2.2015E4	808.48	147.60									
N15	SA	30	3.0401E4	2386.84	435.77	1.764	0.189	-4.808	58	0.000	-2252.92	468.55	-3190.84	-1315.00
	GA	30	3.2654E4	943.07	172.18									

Table 12. The results of T-test for computational times

Group Statistics						Independent Samples Test								
						Levene's Test for Equality of Variances		t-test for Equality of Means						
													95% Confidence Interval of the Difference	
Problem Name	Algorithm	N	Mean	Std. Deviation	Std. Error Mean	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
N01	SA	30	7.94	0.49	0.08	0.472	0.495	20.886	58	0.000	3.48	0.167	3.15	3.82
	GA	30	4.45	0.77	0.14									
N02	SA	30	8.63	0.56	0.10	2.448	0.123	31.351	58	0.000	3.92	0.125	3.67	4.17
	GA	30	4.71	0.38	0.07									
N03	SA	30	8.30	0.58	0.10	0.002	0.967	25.947	58	0.000	3.72	0.143	3.43	4.00
	GA	30	4.58	0.52	0.09									
N04	SA	30	8.18	0.46	0.08	4.196	0.045	33.183	53.266	0.000	3.46	0.104	3.25	3.67
	GA	30	4.71	0.33	0.06									
N05	SA	30	9.74	0.62	0.11	1.794	0.186	27.062	58	0.000	4.15	0.153	3.84	4.46
	GA	30	5.59	0.56	0.10									
N06	SA	30	12.13	0.55	0.10	0.032	0.859	33.870	58	0.000	4.85	0.143	4.57	5.14
	GA	30	7.28	0.55	0.10									
N07	SA	30	14.85	0.49	0.09	8.521	0.005	32.950	46.895	0.000	5.88	0.178	5.52	6.24
	GA	30	8.96	0.84	0.15									
N08	SA	30	14.77	0.29	0.05	7.992	0.006	46.184	42.170	0.000	5.67	0.122	5.42	5.91
	GA	30	9.10	0.60	0.11									
N09	SA	30	15.22	0.37	0.06	25.947	0.000	25.644	35.931	0.000	5.33	0.208	4.91	5.75
	GA	30	9.89	1.07	0.19									
N10	SA	30	13.76	0.81	0.14	2.076	0.155	25.003	58	0.000	4.94	0.197	4.55	5.34
	GA	30	8.81	0.71	0.13									
N11	SA	30	17.59	0.45	0.08	23.214	0.000	30.277	38.306	0.000	6.64	0.219	6.20	7.09
	GA	30	10.90	1.11	0.20									
N12	SA	30	17.52	0.44	0.08	8.505	0.005	42.281	47.228	0.000	6.70	0.158	6.38	7.02
	GA	30	10.82	0.74	0.13									
N13	SA	30	18.92	0.54	0.09	12.995	0.001	35.584	43.416	0.000	7.65	0.215	7.22	8.08
	GA	30	11.27	1.04	0.19									
N14	SA	30	21.02	0.78	0.14	0.430	0.514	30.776	58	0.000	7.15	0.232	6.68	7.61
	GA	30	13.87	1.00	0.18									
N15	SA	30	23.81	0.99	0.18	4.462	0.039	25.436	52.102	0.000	8.03	0.315	7.40	8.67
	GA	30	15.77	1.41	0.25									

According to the results listed in Tables 10-12, it can be seen that SA obtains lower average objective values in 14 out of 15 test problems where in 8 of them, 2 small, 2 medium and 4 large test problems, the results have statistical significance. Therefore, it can be stated that SA relatively outperforms GA in terms of objective values, especially when the sizes of the test problems become larger. In terms of computational times, GA generally performs better and the results of all test problems indicate statistical significance. Finally, it should be mentioned that, the performance of GA improves by allowing longer running time (e.g. more generations or larger population size), but the gradient of this progress is diminishing as the dimensions of the problem increase.

5.5. Sensitivity analysis

The green and environmental concerns are shown in three different aspects for the proposed model: (1) the cost of total CO_2 emissions, (2) variable costs of the fleet which includes fuel consumption cost and (3) the revenue gained by collecting returned goods; where the first two terms should be minimized and the last one must be maximized; In this section, the sensitivity analysis is described which is established in order to figure out the impact of collecting returned goods on the cost of total CO_2 emissions, variable costs of the fleet and the objective function value. The same test problems of Table 8 are considered and solved using the SA algorithm two times; One time when collecting returned goods is allowed (returned goods are available) and the other time while it is not (returned goods are not available). Each test problem is run 5 times in each turn and the solution with the best objective function value is picked to perform the analysis. The results are represented in Table 13. As it can be observed, the cost of total CO_2 emissions is reduced by 6.02%, 16.91% and 10.43% respectively for the small, medium and large-sized problems when the returned goods are not available. In the same way, variable costs of the fleet are reduced by 13.68%, 18.43% and 12.38%. On the contrary, the objective function value is increased by 61.54%, 26.78% and 18.27% for the mentioned categories which is considered as the loss of revenue gained by collecting returned goods while they are not available. In other words, if collecting returned goods is allowed, these percentages represent the reduction in total costs. Eventually, considering the whole test problems, it can be concluded that collecting returned goods causes 11.12% increase in the cost of total CO_2 emissions and 14.89% increase in variable costs of the fleet (including fuel consumption cost), while leads to 35.53% reduction of the total costs. Fig. 14-16 depict the results of Table 13.

Table 13. Sensitivity analysis

	The cost of total CO_2 emissions			**Variable costs of the fleet**			**Objective function (Total costs)**		
Problem Name	With returned goods	Without returned goods	Reduction percentage	With returned goods	Without returned goods	Reduction percentage	With returned goods	Without returned goods	Increase percentage
N01	1352.5	1280.4	5.33%	973.1	854.9	12.14%	2269.5	6511.6	65.14%
N02	1348.5	1254.1	7.00%	962.2	854.9	11.15%	2384.1	6431.7	62.93%
N03	1383.0	1319.6	4.58%	973.8	854.9	12.21%	2131.7	6504.2	67.22%
N04	1508.7	1391.3	7.78%	1142.3	951.3	16.72%	2993.1	6803.7	56.00%
N05	1511.8	1429.6	5.43%	1147.9	951.3	17.12%	3055.9	7013.7	56.42%
Mean%			**6.02%** ▼			**13.68%** ▼			**61.54%** ▲
N06	2789.0	2206.9	20.87%	1791.5	1288.4	28.08%	6160.7	10006	38.42%
N07	2601.2	2313.3	11.06%	1555.8	1445.1	7.11%	7543.4	10338	27.03%
N08	3056.2	2317.7	24.16%	2107.5	1493.1	29.15%	8292.8	10404	20.29%
N09	3026.3	2559.9	15.41%	2051.7	1624.1	20.84%	8501.3	10975	22.53%
N10	3816.1	3316.6	13.08%	2349.3	2185.1	6.98%	10404	13991	25.63%
Mean%			**16.91%** ▼			**18.43%** ▼			**26.78%** ▲
N11	3590.6	2916.9	18.76%	2363.7	1909.8	19.20%	10990	12848	14.46%
N12	4339.4	3806.1	12.28%	2944.3	2468.4	16.16%	11869	15341	22.63%
N13	5145.0	4695.1	8.74%	3510.1	3108.3	11.44%	15610	19300	19.11%
N14	6324.6	5884.6	6.95%	4276.7	3818.1	10.72%	18559	24271	23.53%
N15	8375.6	7920.3	5.43%	6051.1	5783.6	4.42%	29417	33299	11.65%
Mean%			**10.43%** ▼			**12.38%** ▼			**18.27%** ▲
Total Mean%			**11.12%** ▼			**14.89%** ▼			**35.53%** ▲

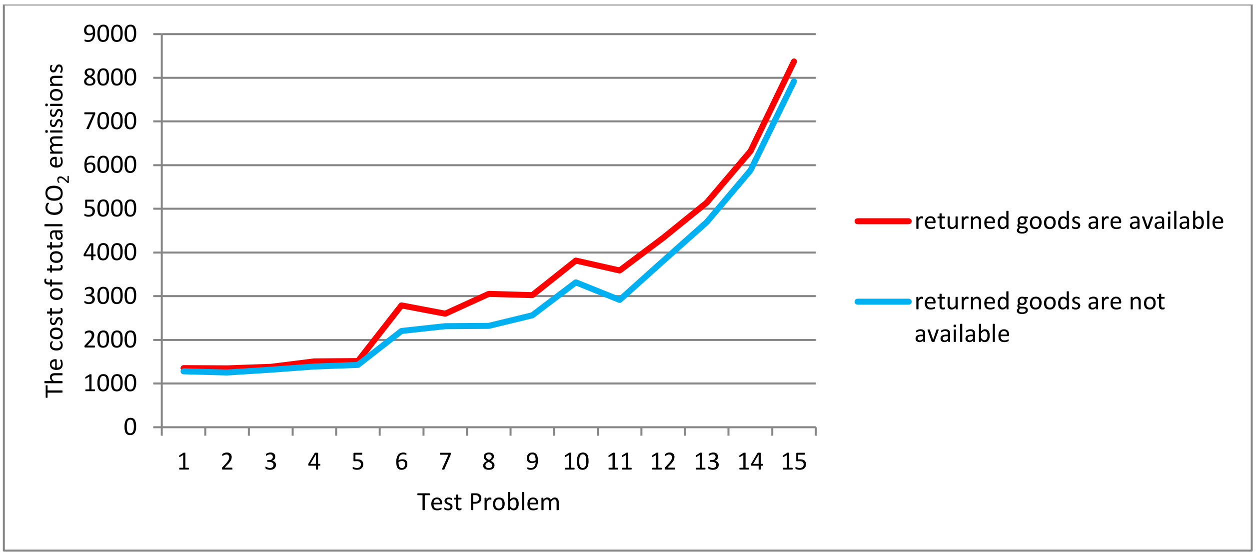

Fig. 14 The effect of collecting returned goods on the cost of total CO_2 emissions

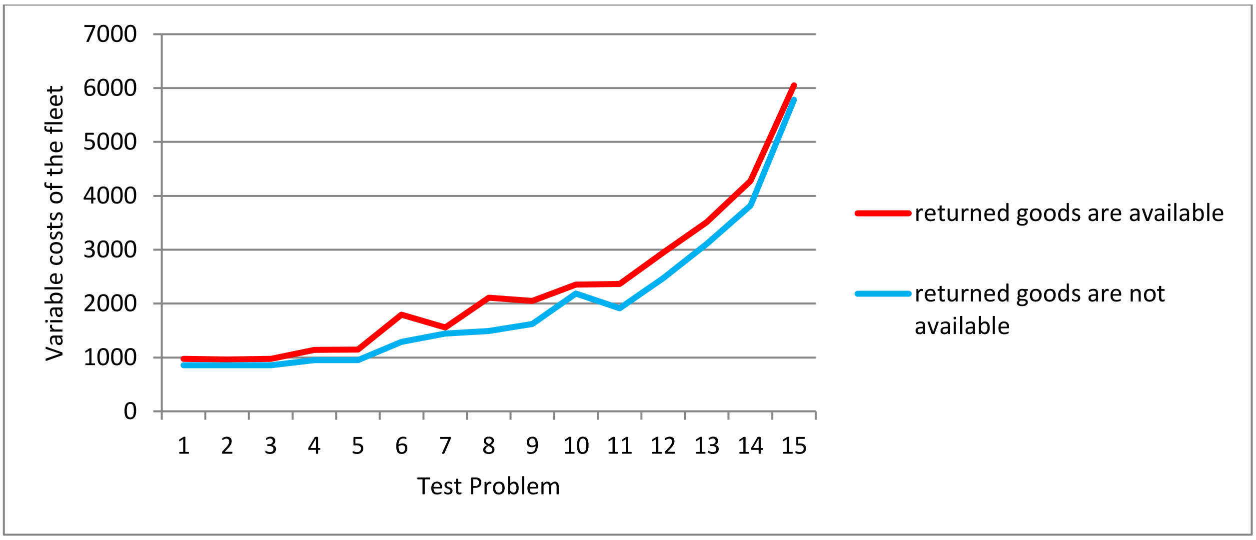

Fig. 15 The effect of collecting returned goods on variable costs of the fleet

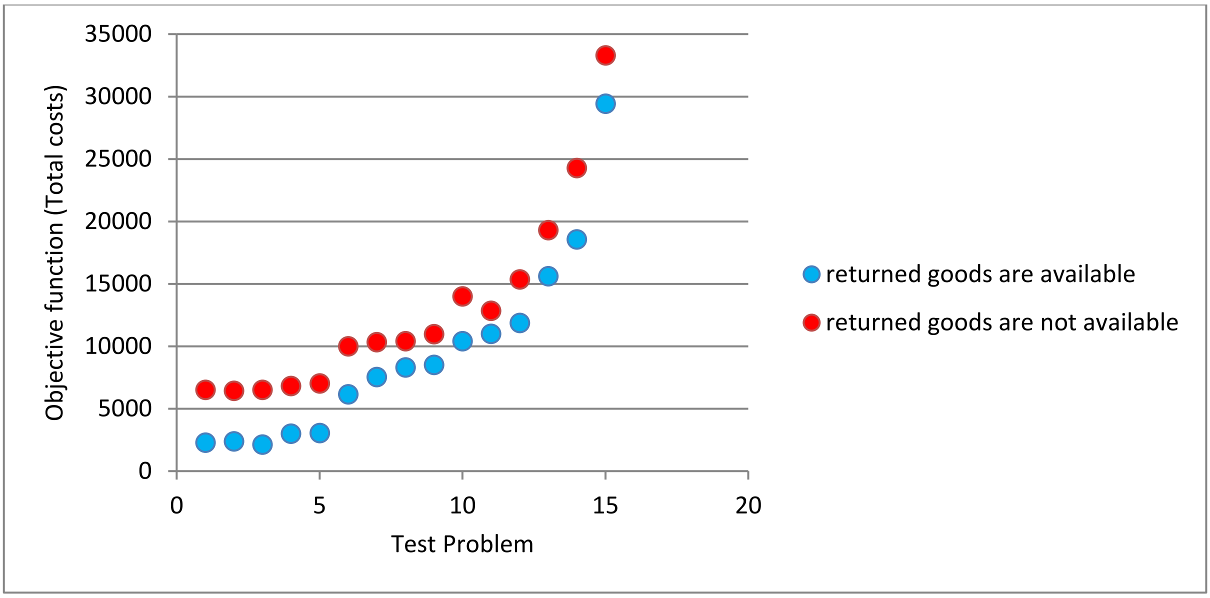

Fig. 16 The effect of collecting returned goods on the objective function value (total costs)

6. Conclusions

Vehicle routing problem is one of the most known and applicable problems in transportation and logistics context. As environmental concerns have become more important over last decade, green VRP has been taken into consideration increasingly. In this study, green vehicle routing and scheduling problem with heterogeneous fleet including reverse logistics in the form of collecting returned goods along with weighted earliness and tardiness costs was considered. The problem was first formulated as a mixed integer non-linear programming model; then has been evaluated in a three-stage procedure using LINGO 9.0. and was shown that it is precise, accurate and well-formulated. Since the problem is classified as NP-hard, in order to find near-optimal solutions for medium and large-sized problems in a reasonable computational time, a simulated annealing and a genetic algorithm was presented. Finally, using a set of defined test problems, performances of the proposed meta-heuristics were evaluated with respect to two considered criteria, solutions quality and computational times. The results achieved by algorithms were compared with each other for all-sized test problems and also with the ones achieved by LINGO for small-sized instances. The conducted analysis indicated that the proposed algorithms have a satisfying performance in achieving high quality near-optimal solutions in a reasonable time. Relatively, SA was better in obtaining higher quality solutions especially for large-sized problems, while GA ran faster for all test problems. Sensitivity analysis indicated that although collecting returned goods causes an 11.12% and 14.89% increase in the cost of total CO_2 emissions and variable costs of the fleet respectively, but can results in a total 35.53% reduction in the objective function value. Collecting returned goods from customers in the framework of reverse logistics which has not been studied in the literature of GVRSP, considering a heterogeneous fleet, the changes in the weight of vehicles throughout their routes and applying earliness and tardiness penalties for demands which are not delivered on time were the main features of the present study. Eventually, for the future studies, considering multi-depot or multi-product situations, condition in which there are temporary warehouses and cross-docks along the routes and also the possibility of supplying each customer with more than a single visit are suggested.

REFERENCES

[1] J. K. Lenstra, A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. Networks, 11(2), (1981). 221–227.

[2] G. B. Dantzig, J. H. Ramser. The truck dispatching problem. Management Science, 6(1), (1959), 80-91.

[3] The International Energy Agency, CO2 emissions from fuel combustion highlights (2015 Edition), 2015, http://www.iea.org/publications/freepublications/publication/CO2EmissionsFromFuelCombustionHighlights2015.pdf (accessed 06.02.16).

[4] J. M. Bloemhof-Ruwaard, P. Van Beek, L. Hordijk, L. N. Van Wassenhove. Interactions between operational research and environmental management. European Journal of Operational Research, 85(2), (1995), 229–243.

[5] S. E. Daniel, D. C. Diakoulaki, C.P. Pappis, Operations research and environmental planning. European Journal of Operational Research, 102(2), (1997), 248–263.

[6] I. Kara, B.Y. Kara, M.K. Yetis. Energy minimizing vehicle routing problem. Combinatorial Optimization and Applications, 4616, (2007), 62–71.

[7] S. Erdoğan, E. Miller-Hooks. A green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review, 48(1), (2012), 100–114.

[8] D. R. Gaur, A. Mudgal, R. R. Singh. Routing vehicles to minimize fuel consumption. Operations Research Letters, 41(6), (2013), 576–580.

[9] C. Lin, K.L Choy, G.T.S. Ho, S.H Chung, H.Y Lam. Survey of green vehicle routing problem: past and future trends. Expert Systems with Applications, 41(4)(p1), (2014), 1118–1138.

[10] S. M. J. Mirzapour Al-e-hashem, Y. Rekik. Multi-product multi-period inventory routing problem with a transshipment option: A green approach. International Journal of Production Economics, 157, (2014), 80-88

[11] I. M. Coelho, P. L. A. Munhoz, L. S. Ochi, M. J. F. Souza, C. Bentes, R. Farias. An integrated CPU–GPU heuristic inspired on variable neighbourhood search for the single vehicle routing problem with deliveries and selective pickups. International Journal of Production Research, 54(4), (2016), 945-962.

[12] M. Soysal. Closed-loop inventory routing problem for returnable transport items. Transportation Research Part D: Transport and Environment, 48, (2016), 31–45.

[13] A. Franceschetti, D. Honhon, T.V. Woensel, G. Laporte, T. Bektas. The time-dependent pollution-routing problem. Transportation Research Part B: Methodological, 56, (2013), 265–293.

[14] Y. J. Kwon, Y. J. Choi, D. H. Lee. Heterogeneous fixed fleet vehicle routing considering carbon emission. Transportation Research Part D: Transport and Environment, 23, (2013), 81–89.

[15] T. Bektas, G. Laporte. The Pollution-Routing Problem. Transportation Research Part B, 45(8), (2011), 1232–1250.

[16] E. Demir, T. Bektaş, G. Laporte. An adaptive large neighborhood search heuristic for the Pollution-Routing Problem. European Journal of Operational Research, 223(2), (2012), 346–359.

[17] E. Demir, T. Bektaş, G. Laporte. The bi-objective Pollution-Routing Problem. European Journal of Operational Research, 232(3), (2014), 464–478.

[18] Ç. Koç, T. Bektaş, O. Jabali, G. Laporte. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. Computers & Operations Research, 64, (2015), 11.27.

[19] M.A. Figliozzi. Vehicle routing problem for emissions minimization. Transportation Research Record: Journal of the Transportation Research Board, 2197, (2010) 1–7.

[20] Y. Xiao, A. Konak. A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness. Applied Soft Computing, 34, (2015), 372–388.

[21] C. Malandraki, M. S. Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. Transportation Science, 26(3), (1992), 185–200.

[22] E. Demir, T. Bektaş, G. Laporte. A review of recent research on green road freight transportation. European Journal of Operational Research, 237(3), (2014), 775–793.

[23] A. J. Hickman, D. Hassel, R. Joumard, Z. Samaras, S. Sorenson. Methodology for calculating transportation emissions and energy consumption. Transport research laboratory, 22, (1999), 69–73.

[24] The ministry of land, infrastructure, transport and tourism of Japan website, http://www.mlit.go.jp/common/000037099.pdf (accessed 02.08.16).

[25] Y. Xiao, Q. Zhao, I. Kaku, Y. Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. Computers & Operations Research, 39(7), (2012), 1419–1431.

[26] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. Optimization by simulated annealing. Science, New Series, 220(4598), (1983), 671-680.

[27] Holland, J. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. Ann Arbor, MI: University of Michigan Press. (1975).

[28] G. Taguchi. Introduction to quality engineering. Asian Productivity Engineering, Quality Resources, White Plains, New York (1986).

[29] J. U. Sun. A Taguchi approach to parameter setting in a genetic algorithm for general job shop scheduling problem. IEMS, 6(2), (2007), 119-124.

[30] Y. Xiao, A. Konak. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. Transportation Research Part E: Logistics and Transportation Review, 88, (2016), 146-166.

[31] Y. Xiao, A. Konak. A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem. Journal of Cleaner Production, 167, (2017), 1450-1463.

[32] R. Raeesi, K. Zografos. Solving the time-and-load dependent green vehicle routing and scheduling problem on real road networks. VeRoLog 2016, (2016).

[33] D. Zhang, X. Wang, S. Li, N. Ni, Z. Zhang. Joint optimization of green vehicle scheduling and routing problem with time-varying speeds. PLOS ONE 13(2): e0192000, (2018).

[34] M. M. Musavi, R. Tavakkoli-Moghaddam, F. Rayat. A bi-objective green truck routing and scheduling problem in a cross dock with the learning effect. Iranian Journal of Operations Research, 8(1), (2017), 2-14.

[35] N. Nabil, H. A. Farouk, K. S. El-Kilany. Green vehicle routing and scheduling problem with optimized travel speed. IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (2017).

[36] K. N. Androutsopoulos, K. G. Zografos. An integrated modelling approach for the bicriterion vehicle routing and scheduling problem with environmental considerations, 82, (2017), 180-209.

[37] M. Rabbani, S. Aghamohammadi Bosjin, R. Yazdanparast, N. Akbarian Saravi. A stochastic time-dependent green capacitated vehicle routing and scheduling problem with time window, resiliency and reliability: a case study. Decision Science Letters, 7, (2018), 381–394.

[38] M. R. S. Matos, Y. Frota, L. S. Ochi. Green Vehicle Routing and Scheduling Problem with Split Delivery. Electronic Notes in Discrete Mathematics, 69, (2018), 13-20.