Environ Model Assess (2015) 20:273–284
DOI 10.1007/s10666-014-9434-2

Reduction of CO_2 Emissions in Cumulative Multi-Trip Vehicle Routing Problems with Limited Duration

Didem Cinar · Konstantinos Gakis · Panos M. Pardalos

Received: 15 April 2014 / Accepted: 24 October 2014 / Published online: 12 November 2014

Abstract In recent years, as a result of the increase in environmental problems, green logistics has become a focus of interest by researchers, governments, policy makers, and investors. In this study, a cumulative multi-trip vehicle routing problem with limited duration (CumMTVRP-LD) is modelled by taking into account the reduction of CO_2 emissions. In classical vehicle routing problems (VRP), each vehicle can perform only one trip. Because of the high investment costs of additional vehicles, organizations allow the vehicles to perform multiple trips as in multi-trip vehicle routing problems (MTVRP), which reflects the real requirements better than the classical VRP. This study contributes to the literature by using a mixed integer programming (MIP) formulation and a simulated annealing (SA) based solution methodology for CumMTVRP-LD, which considers the minimization of fuel consumption as the objective function. According to preliminary computational results using benchmark problems in the literature, the proposed methodology obtained promising results in terms of solution quality and computational time.

Keywords CO_2 Emissions · Multi-trip vehicle routing problem · Mixed integer programming · Heuristic approaches · Simulated annealing

1 Introduction

Along with the technological and economic development, the effect of the vehicles on environment have attracted attention of researchers and governments. Environmental, social and political concerns result in pressure on companies to decrease CO_2 emissions, which is one of the main greenhouse gases that cause global warming [40]. Since such emissions cause environmental problems and deterioration of human health and social welfare, environmental and social factors should be considered alongside with the efficiency of transportation [29]. Organizations want to decrease CO_2 emissions not only for business ethics purposes effecting the social impact but also for economic reasons.

According to the International Energy Agency data for 2011 [21], although the USA accounts for only 4.5 % of the world population, it accounts for 17 % of global CO_2 emissions. Electric power and transportation sectors account for approximately three quarters of all the CO_2 emissions in the USA and it seems unlikely to change in the next three decades [33]. The transportation sector in the USA accounts for 33 % of global transportation CO_2 emissions and about 7 % of global CO_2 emissions [37]. Therefore, CO_2 emissions reduction strategies developed for the US transportation sector may have a remarkable impact on global CO_2 and greenhouse gas emissions. Since 1990, the

D. Cinar
Department of Industrial Engineering, Faculty of Management, Istanbul Technical University, Istanbul, Turkey
e-mail: didemcin@gmail.com

D. Cinar · K. Gakis · P. M. Pardalos (✉)
Department of Industrial and Systems Engineering, Faculty of Engineering, University of Florida, Gainesville, USA
e-mail: pardalos@ufl.edu

K. Gakis
e-mail: kgakis@ufl.edu

P. M. Pardalos
Laboratory of algorithms and technologies for network analysis (LATNA), National Research University Higher School of Economics, Moscow, Russia

largest increase in the greenhouse gas emissions is from medium and heavy-duty trucks, which are especially used for transportation (77 % increase has been observed for medium and heavy-duty trucks, which is three times larger than the rate of light-duty vehicles) [37]. The US Department of Transportation has been investigating strategies "to improve transportation system efficiency by improving transportation operations and reducing energy use and greenhouse gas emissions associated with a given unit of passenger or freight travel" [37].

Since fuel consumption is one of the most important causes for CO_2 emissions, a multi-trip vehicle routing problem (MTVRP) to minimize fuel consumption as its objective function is considered in this study. The MTVRP differs from the classical vehicle routing problem (VRP) by allowing the assignment of multiple trips with the same vehicle. Most of the VRP studies have taken into account the objectives being linear functions of distance [26], such as total travel time or total transportation cost. Andress et al. [3] grouped the strategies to reduce the greenhouse gas emissions from transportation fuels into three main categories: improving the engine efficiency, using the low carbon fuels and reducing the travel distance. In this study, another aspect on reducing the travel distance to minimize the CO_2 emissions is used. The fuel consumption of a vehicle depends on the transportation plan and the flow of moved loads. A reduction of carbon emissions can be provided by the reduction of fuel consumption which is resulted with the decrease in the weight of vehicles. The weight of a vehicle includes both its own mass without load and the weight of load. The load flow of a vehicle does not depend on the distance. Therefore, to reduce fuel consumption, in addition to the distance, flow on the related arc should be included as another indicator of the cost. Load flow can be represented as a step function. In delivery case, a vehicle on a road carries the demand of all customers who will be served by the same vehicle in that tour and the moved load decreases as a step function while the customers are visited. The reverse is valid for the collection case: the load of a vehicle can be represented as an increasing step function towards the end of the tour. That is why Kara et al. [22] named routing problems with flow based cost functions as *cumulative routing problems*. Since also a time limit restriction for each vehicle is taken into account, the problem investigated in this study is referred to cumulative multi-trip vehicle routing problem with limited duration (CumMTVRP-LD). A mixed integer programming (MIP) formulation for CumMTVRP-LD to minimize the fuel consumption within a planning time horizon and a solution methodology are proposed. The total fuel consumption is determined according to the distance, the load, and the type of the vehicles.

The rest of the paper is organized as follows. The next section gives a brief explanation of CumMTVRP-LD. Section 3 provides the literature review on MTVRP. The proposed MIP formulation is presented in Section 4, while the SA-based methodology is explained in Section 5. The preliminary computational results are reported and analysed in Section 6. Finally, concluding remarks and further suggestions are given in Section 7.

2 Problem Statement

A CumMTVRP-LD can be represented on a directed graph $G = (V, A)$ where $V = \{0, 1, \ldots, N\}$ is the set of vertices (nodes) defined as the customers and $A = \{(i, j) \mid i, j \in V, i \neq j\}$ is the set of arcs. Vertex 0 refers to the depot where each vehicle starts and finishes the corresponding route. There is a set $M = \{1, \ldots, K\}$ referring the heterogeneous fleet, which includes K vehicles having different capacity $Q_k (k \in M)$ and fuel consumption rates. The distance from customer i to customer j is represented by $d_{ij}, i, j = 0, 1, \ldots, N$. d_{0i} and d_{i0} represent the distances from depot to customer i and from customer i to depot, respectively. Each customer has a demand of $c_i, i = 1, \ldots, N$. In each demand point, the service time is given as $p_i, i = 1, \ldots, N$, which includes only the unloading time. Reloading is not considered in this study. The whole delivery process of a vehicle should end before a predetermined time limit D for the duration of the trip, which includes both transportation and loading time; overtime is not allowed.

CumMTVRP-LD deals with the determination of optimum route set and optimum assignment of these routes to the vehicles within a predetermined time horizon. Multiple trips can be assigned to the same vehicle. Total demand of the customers in the same tour cannot exceed the vehicle's capacity. Each customer should be visited once and the demand of each customer should be satisfied by only one vehicle. Moreover, an instance is feasible if it satisfies $(d_{0i} + d_{i0})/v \leq D$ and $c_i \leq Q_k$ for all $i \in V \setminus \{0\}$ and $k \in M$, where v represents the average speed of the vehicles.

CO_2 emissions are related with the fuel consumption. Several emission models have been proposed to compute fuel consumption [28]. Kopfer and Kopfer [24] and Kara et al. [22] expressed fuel consumption as a function of the distance and the weight of the vehicle. Although the model and solution methodology are proposed considering only the delivery case in this study, they can easily be applied for collection case without loss of generality.

3 Literature Review

In the literature for optimization, a number of different formulations and an even greater number of algorithms have been proposed for VRP [31]. Although the MTVRP has more realistic assumptions, it has not been studied as widely as the VRP.

As a generalization of the travelling salesman problem, almost all variations of the vehicle routing problem are NP-hard and cannot be solved exactly by polynomial time algorithms [25]. Since MTVRP is a variant of simpler problem, i.e. VRP, it is NP-hard in a strong sense. Therefore, the exact optimization methods may be difficult to solve the large instances in acceptable CPU times.

The first known study on the MTVRP was performed by Fleischmann in 1990 [16]. A bin packing problem heuristic was used to assign the routes on the vehicles. The most widely used metaheuristic approach in the MTVRP literature is tabu search algorithm. Taillard et al. [15] used a methodology based on tabu search to find various solutions for classical VRP and combined these routes to construct the daily plans of the vehicle by a heuristic approach. A data set was generated to evaluate the performance of the proposed methodology. This data set has been used by many studies for comparison of algorithms. Brandão and Mercer [8, 9] used a tabu search algorithm with nearest neighbor search and insertion algorithms for different types of objectives (minimization of transportation cost and minimization of total travel time). Alonso et al. [2] developed a formulation for a site-dependent MTVRP in which not all vehicles can visit all customers. They used a tabu search algorithm for various VRPs and compared it with the algorithms from the literature. Promising results were obtained within reasonable computational time. Nguyen et al. [34] proposed a formulation and developed a tabu search approach for time-dependent multi-zone MTVRP with time windows.

Petch and Salhi [36] proposed a multi-phase algorithm for a MTVRP minimizing the maximum overtime. They used the saving algorithm developed by Yellow [43] to generate a solution set and a MTVRP solution is obtained by the packing heuristic applied on the set. 2-opt and 3-opt heuristics were performed to improve the solutions. Olivera and Viera [35] proposed a set covering formulation and an adaptive memory procedure for MTVRP. They inferred that the optimal solution of single tour VRP is not a good solution for the problem with low-capacity vehicles and long service times. Salhi and Petch [38] developed a genetic algorithm approach to minimize the maximum driver overtime. Huang and Lee [19] considered both the MTVRP and a distribution center location problem and proposed a mathematical programming model for the MTVRP. They developed a three-phase approach including SA algorithm to minimize the total cost, which consists of both transportation cost and activated vehicle costs. Cattaruzza et al. [11] studied the MTVRP in which low-capacity vehicles are used. They developed a hybrid genetic algorithm to minimize the total travel distance and obtained better results than the algorithms in the literature. Azi et al. [6] used an adaptive neighborhood search approach for MTVRP in which the demands of all customers cannot be satisfied. They developed operators at different levels as customer, route and work day for the maximization of the number of served customers and minimization of total distance. According to the computational results, the proposed operators obtained better performance than the ones at studies using operators only with customer level. A detailed explanation of the fundamental heuristic algorithms developed for MTVRP and the comparison of their performance can be found in Şen and Bülbül [14].

Besides the above studies that rely on heuristic algorithms for MTVRP, a few studies dealt with exact algorithms. Azi et al. [5] used a branch-and-price approach to maximize total revenue and minimize total distance for MTVRP with a time window and a revenue associated with each customer. The lower bounds were computed by the linear programming relaxation of a set packing formulation and the pricing subproblems were handled as elementary shortest path problems with resource constraints. Mingozzi et al. [32] developed two set partitioning-like formulations and an exact algorithm for the MTVRP. The computational experiments showed that instances with up to 120 customers can be solved with the proposed algorithm.

Azi et al. [4] presented an exact two-phase algorithm, which is based on an elementary shortest path algorithm with resource constraints, to solve MTVRP with time windows for single-vehicle case. Azi et al. [5] advanced the study for the same problem with multiple vehicles. Macedo et al. [30] developed an iterative exact algorithm based on a pseudo-polynomial network flow model for MTVRP with time windows. In all these studies, there was no restriction to serve all customers. Hernandez et al. [18] proposed an exact two-phase algorithm for multi-trip vehicle routing problem with time windows and limited duration. According to the computational results, Hernandez et al. [18] obtained lower computational times for the instances having up to 40 customers than the algorithms developed by Azi et al. [4] and Macedo et al. [30].

The studies mentioned above are classified in Table 1 according to their objective functions. Minimization of transportation cost is the most widely used objective in MTVRP literature. To the best of our knowledge, the above studies have not considered CumMTVRP-LD which has the objective of fuel consumption minimization as seen in Table 1. A detailed literature survey on various green VRPs can be found in Lin et al. [27] where the CumMTVRP-LD is not mentioned as a problem studied with environmental concerns.

This study contributes to the literature with a new mathematical programming formulation of the CumMTVRP-LD.

Table 1 The classification of MTVRP studies with respect to objective function

Study	O1	O2	O3	O4	O5	O6
Azi et al. [6]	✓	✓				
Azi et al. [5]		✓			✓	
Taillard et al. [15]		✓				
Cattaruzza et al. [11]			✓			
Brandão and Mercer [9]			✓			
Mingozzi et al. [32]				✓		
Nguyen et al. [34]				✓		
Alonso et al. [2]				✓		
Olivera and Viera [35]				✓		
Brandão and Mercer [8]				✓		
Salhi and Petch [38]						✓
Petch and Salhi [36]						✓

O1: maximization of the number of served customers
O2: minimization of total distance
O3: minimization of total travel time
O4: minimization of total transportation cost
O5: maximization of revenue
O6: minimization of overtime

Besides the distance, the objective function optimized in this study relates with the load, which requires additional decision variables in the mathematical formulation. A methodology, which combines several heuristics with a well-known metaheuristic algorithm, that of SA, is proposed to find good solutions for the CumMTVRP-LD in a reasonable time. SA is a probabilistic approximation algorithm which improves a solution by walking randomly in the solution space to avoid trapping in local optima. Unlike the other single-solution-based metaheuristics, it exploits worse solutions to find the global optimum. Since it is able to produce good solutions in short times for various optimization problems and easy-to-implement, SA is used in the methodology developed for CumMTVRP-LD. Moreover, some preliminary computational results obtained from benchmark instances are reported and analysed.

4 A MIP Formulation

The model of Hernandez et al. [18] although similar to the one proposed in this paper to most of its aspects, it differs essentially in two aspects: (1) with respect to the constraint of the duration limit, which in our case includes the total trip duration, i.e. it includes all loading times and (2) the objective function includes costs associated with fuel consumption, which in turn depends on the load carried, thus adding a degree of complexity to the model. Two secondary differences are (1) Hernandez et al. [18] consider time windows for the delivery service and (2) the proposed model allows for variable capacity of the trucks.

It should be underscored that the addition of load-dependent fuel consumption in the objective function as well as the inclusion of loading times in the constraint for duration constitutes a major change in the model, as they essentially inhibit the use of the exact algorithm by Hernandez et al. [18]. The later depends on the establishment of dominance relations, which in their case depend only on the structural characteristics of the network (costs). Such an approach is not immediately applicable to our model as the introduction of the load into the objective function renders the dominance established invalid.

The notation used in the proposed formulation is given as follows.

Indices:

$i, j:$ Customers ($i, j = 0, 1, \ldots, N, N+1$ where 0 and $N+1$ refer to the depot).
$k:$ Vehicles ($k = 1, \ldots, K$)
$t:$ Trips ($t = 1, \ldots, T$)

Parameters:

$N:$ The number of customers
$K:$ The number of vehicles
$T:$ The number of trips
$D:$ The time limit that a vehicle can operate
$v:$ Average speed of a vehicle
$p_j:$ Service time for city j ($p_0 = p_{N+1} = 0$)
$d_{ij}:$ Distance between city i and city j
$c_i:$ Demand of city i
$Q_k:$ Capacity of vehicle k
$a_k:$ Fuel consumption of empty vehicle k per kilometer
$b_k:$ Fuel consumption of vehicle k per ton and kilometer

Variables:

$$x_{ijk}^t = \begin{cases} 1 & \text{if vehicle } k \text{ visits city } j \text{ immediately after city } i \text{ on trip } t \\ 0 & \text{otherwise} \end{cases}$$

$$z_k^t = \begin{cases} 1 & \text{if vehicle } k \text{ performs trip } t \\ 0 & \text{otherwise} \end{cases}$$

$q_{ijk} =$ total load transported from city i to city j by vehicle k

The fuel consumption considered as objective function is handled as in Kopfer and Kopfer [24]. Vehicle categories are defined by the capacities and fuel consumptions. Besides the assignment variables (x_{ijk}^t) of customers to vehicles and load variables (q_{ijk}), assignment variables (z_k^t)

of trips to vehicles are used. The proposed formulation for the CumMTVRP-LD is given as follows.

$$\min \sum_{i=0}^{N}\sum_{j=1}^{N+1}\sum_{k=1}^{K} d_{ij}\left(a_k \sum_{t=1}^{T} x_{ijk}^t + b_k \cdot q_{ijk}\right) \tag{1}$$

s.t.

$$\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{i=0;i\neq j}^{N} x_{ijk}^t = 1 \qquad j = 1,\ldots,N \tag{2}$$

$$\sum_{i=0,i\neq j}^{N} x_{ijk}^t = \sum_{i=1,i\neq j}^{N+1} x_{jik}^t \qquad j = 1,\ldots,N, \forall k, \forall t \tag{3}$$

$$\sum_{i=1,i\neq j}^{N}\sum_{j=1}^{N} x_{ijk}^t \leq (N-1)\cdot z_k^t \qquad \forall k, \forall t \tag{4}$$

$$\sum_{j=1}^{N} x_{0jk}^t = z_k^t \qquad \forall k, \forall t \tag{5}$$

$$\sum_{i=1}^{N} x_{i,N+1,k}^t = z_k^t \qquad \forall k, \forall t \tag{6}$$

$$\sum_{i=0,i\neq j}^{N} q_{ijk} - \sum_{i=1,i\neq j}^{N+1} q_{jik} = c_j \sum_{t=1}^{T}\sum_{i=0,i\neq j}^{N} x_{ijk}^t \qquad j = 1,\ldots,N, \forall k \tag{7}$$

$$q_{ijk} \leq Q_k \sum_{t=1}^{T} x_{ijk}^t \qquad i = 0,\ldots,N,\ j = 1,\ldots,N, \forall k \tag{8}$$

$$\sum_{t=0}^{T}\sum_{i=0}^{N}\sum_{j=1}^{N+1}\left(d_{ij}/v + p_j\right)\cdot x_{ijk}^t \leq D \qquad \forall k \tag{9}$$

$$x_{ijk}^t \in \{0,1\} \qquad i, j = 1,\ldots,N, \forall k, \forall t \tag{10}$$

$$z_k^t \in \{0,1\} \qquad \forall k, \forall t \tag{11}$$

$$q_{ijk} \geq 0 \qquad i = 0,\ldots N,\ j = 1,\ldots N+1, \forall k \tag{12}$$

The objective function (1) minimizes the total fuel consumption, which is a function of distance and load. Constraints (2) ensure that each customer should be visited. Constraints (3) satisfy flow conservation, i.e. if an arrival occurs to customer i then a departure should be performed. Constraints (4) guarantee that an inactive vehicle cannot visit any customer. Constraints Eqs. (5) and (6) satisfy that each tour should start and end at the depot. $(N + 1)$ is an artificial ending depot while 0 is the starting depot. Constraints (7) are subtour elimination constraints, which also satisfy the load flow for each road. The amount of a load, which a vehicle delivers to a customer, can be at most the demand of the related customer. Constraints (8) ensure that total load of a vehicle cannot exceed the capacity of the vehicle. Constraints (9) satisfy the overtime restriction where the sum of the travel time between customers and service time for all customers served by the same vehicle cannot be greater than the given time limit. Service time for depot is taken as 0. Constraints (10–12) are sign restrictions for the decision variables.

5 Heuristic Approaches

In this study, a hybrid methodology, which is a combination of heuristic and metaheuristic approaches, is proposed to obtain good results for CumMTVRP-LD. A solution is represented by an array giving the order of the cities in the tours. The tours in a solution are split by zeros, which can be considered as the depot. A sample solution and its corresponding visual representation for a small instance is given at Fig. 1.

The proposed methodology includes three main phases: (a) initial solution generation, (b) finding the tours and (c) assignment of the tours to the vehicles. An initial solution is generated by the saving algorithm developed by Clarke and Wright [13] for classical VRP. It is an effective constructive heuristic for VRP that proceeds adding a customer to a tour at each iteration. At the beginning of the algorithm, each vertex is assigned to different truck. Then a

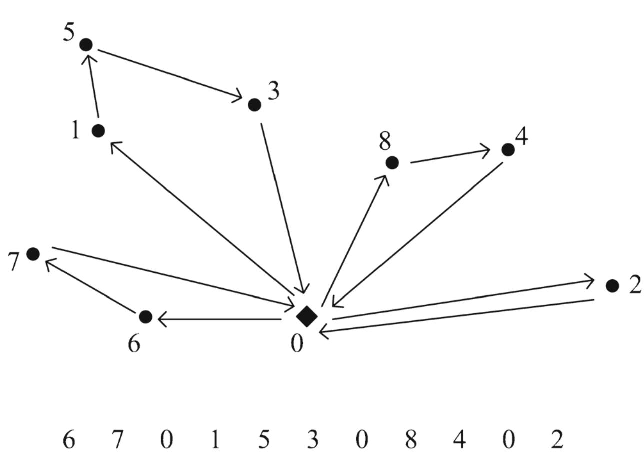

Fig. 1 A sample solution representation

saving value is found for each pair of vertex $(i, j), i < j$ and $i = 0, 1, \ldots, N$. This saving refers the cost of assigning i and j consecutively for the same vehicle rather than assigning them to different vehicles. At the beginning of the algorithm, each vertex i assigned a vehicle separately from the others. Then saving is computed as the following for vertex pair (i, j):

$$d_{0i} + d_{0j} - d_{ij} \tag{13}$$

where $d_{ij} = d_{ji}$ for all $i, j \in V$. After all savings are calculated for each vertex pair, they are sorted in nonincreasing order. A vertex pair (i, j) having the largest saving is assigned first if they are not both assigned already on the same machine and the capacity and time constraints of the vehicle are not violated [13]. Only the first and/or the last vertices are considered to assign consecutively.

After generating an initial solution, a metaheuristic approach is used for the construction of routes where it is assumed that the fleet consists of the same type of vehicles. In this phase, a SA approach is applied. SA is a probabilistic approach that aims to find the global optimum of a cost function that may have several local optima [7]. It was inspired from the physical annealing process of solid matter. Since the atoms remove randomly as a result of heating, the solid is cooled too slowly to reach the minimum energy configuration of its atoms. A very slow cooling process allows atoms to reach a thermal equilibrium [10]. In optimization problems, the state of the solid refers to the feasible solution while the minimum energy configuration refers to the optimal solution [42]. For a mathematical background of SA, readers are referred to Bertsimas and Tsitsiklis [7] and Alizamir et al.[1]. After getting a tour combination by SA, a set-packing heuristic approach is applied to assign the routes to the vehicles.

In SA, a neighbor is generated by using one-to-one exchange, reverse and delete-insert operators, which have detailed explanation in [23]. One-to-one exchange operator swaps two randomly selected customers. Reverse operators changes the order of the customers on a randomly selected substring. Delete-insert operation deletes a randomly selected operation and inserts it in another randomly selected position. Zeros (depot) can be selected in every operation. As a result of an operation, if two zeros come alongside, then one of them is deleted. The operators are illustrated in Fig. 2. The operator performed in each iteration is decided randomly. If SA starts with a large initial temperature, it oscillates between solutions at the very beginning of the algorithm that may even not converge to any local solution. If it starts with small temperature and decreases slowly, the probability of jumping to a new solution to escape local optima gets very small. So a reliable initial temperature should be defined. In this study, different parameters are tried, and it is observed that the initial temperature and the temperature factor proposed in the literature (see [23]) gets better performance. However, while the iterations proceed, escaping from the local optima is getting hard because the probability of choosing a worse neighbor is getting smaller. Therefore, an additional criterion is added in the algorithm, which considers the number of subsequent iterations having the same objective value. If this number exceeds a predetermined limit and no neighbor with a better objective value is found, then a neighbor having worse objective value is replaced with the current solution. This procedure can be seen as a perturbation of the current solution. The pseudo-code of the SA used in this study is given in Algorithm 1.

Algorithm 1 Simulated Annealing

```
input:An initial solution(current_solution)
begin
initialize temperature, min_temp, max_count andcount ← 0
while temperature > min_temp do
    generateneighbor
    if neighbor(total_fuel) < current_solution(total_fuel) then
        current_solution ← neighbor
        count ← 0
    else
        generate random number r
        update temperature and probability
        if r < probability or count > max_count then
            current_solution ← neighbor
            count ← 0
        else
            count ← count + 1
        end if
    end if
end while
get best_solution
end
output:A tour combination
```

6 Computational Results

Since Solomon's VRP with time window instances [39] are widely used in MTVRP studies in the literature, they are also modified and used in this study to determine the performance of the proposed formulation and methodology for CumMTVRP-LD. Instances have been grouped according to the location of the customers (r: random, c: clustered, rc: mixed) and the width of the scheduling horizon (1: short horizon, 2: long horizon). As in Azi et al. [6], only the instances with long horizon (r2, c2 and rc2) are used for computational experiments because a short horizon does not allow a significant number of routes to be sequenced

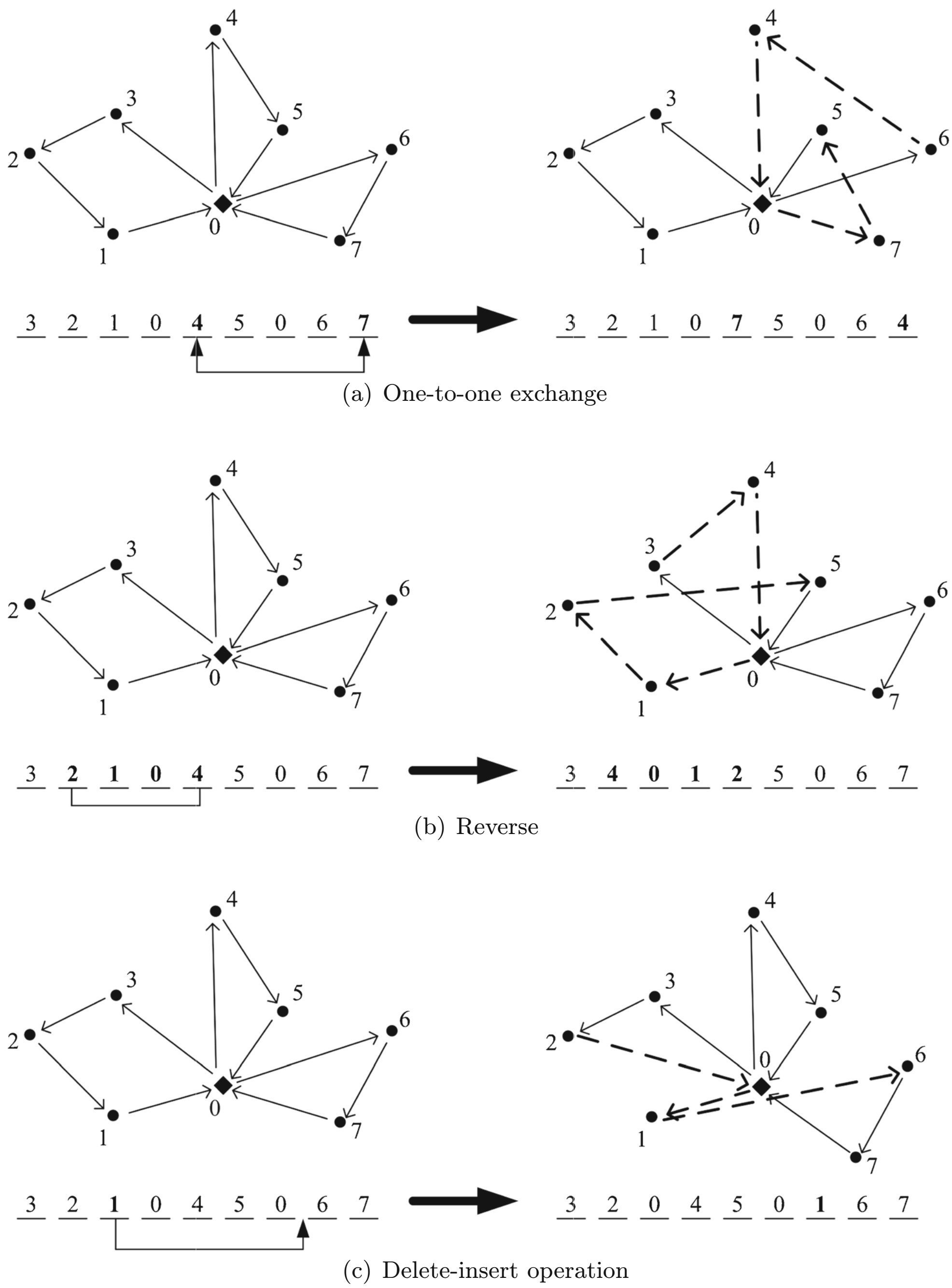

Fig. 2 Operators used to generate neighbors

to form a work day. Since an exact algorithm cannot solve large-size instances in a reasonable time, instances with 10 and 20 customers are generated from Solomon's data set that includes 100 customers. Ten instances are generated for each instance group (r2, c2 and rc2) and size (10 and 20 customers). The instances are named as "instance group_number of customer_instance number". For example, r2_10_5 refers to the fifth instance of the randomly generated instance group having ten customers. Some modifications are performed on the original instances to fit our problem. Service time for each customer is fixed to 10. Time limit is fixed as 300 to get a tight time constraint. Without loss of generality, the average velocity of the vehicles is fixed as 1 for each instance. The fuel consumption rates are obtained from Kopfer and Kopfer [24]. The fleet is homogeneous, i.e. the vehicles in the fleet have same features such as same capacity and fuel consumption rates. Fuel consumption of an empty vehicle per kilometer is 26 while fuel consumption per ton and kilometer is 0.36 for each vehicle. The capacity of each vehicle is 50.

Firstly, the effect of the objective of total fuel consumption minimization is investigated. The results obtained from the MIP model are given on Table 2. The first column represents the name of the instances. The second column up to the forth give the optimum fuel consumption, total travel distance and the number of vehicles obtained by the proposed MIP model with the objective of minimizing total fuel consumption, respectively. The fifth column up to the seventh one represent the same attributes for the objective of minimizing total travel distance. The last two columns include the percentage of decrease for the fuel consumption and increase for the distance in the case of fuel consumption minimization. According to the computational results, up to 10 % fuel consumption reduction has been achieved by minimizing the total fuel consumption without increase in total distance and the number of vehicles. For 18 instances, no increase in distance is observed.

The performance of the solution methodology is investigated by comparing the results with the exact solutions of developed MIP. The parameters of SA are determined by the literature (see [23]) and computational experiments performed during this study. Initial temperature is taken as 20 while the minimum temperature, which is used as a termination condition is 0.1. Temperature factor is taken as 0.95 as in [23] where $T_{i+1} = 0.95T_i$. T_{i+1} is the temperature at iteration $i + 1$ and calculated by multiplying the preceding temperature with 0.95. The probabilities are 0.4, 0.3 and 0.3 for one-to-one exchange, reverse and delete-insert, respectively.

Table 2 Minimization of total fuel consumption (O1) vs. minimization of total travel distance (O2)

	O1			O2			Fuel	Dist.
	Fuel	Dist.	#vhc	Fuel	Dist.	#vhc	Dec.	Inc.
c2_10_1	6643.99	196.46	1	7108.52	196.46	1	6.53	0.00
c2_10_2	5486.09	167.34	1	5842.56	167.34	1	6.10	0.00
c2_10_3	13,542.74	396.93	2	13,618.52	396.93	2	0.56	0.00
c2_10_4	10,348.78	309.11	2	10,535.24	303.43	2	1.77	1.87
c2_10_5	6494.84	194.68	1	6494.84	194.68	1	0.00	0.00
c2_10_6	11,396.71	335.65	2	11,574.34	333.69	2	1.53	0.59
c2_10_7	12,606.8	376.47	2	12,698.2	376.47	2	0.72	0.00
c2_10_8	10,279.76	310.64	2	10,387.44	310.17	2	1.04	0.15
c2_10_9	7406.08	228.45	2	8203.64	228.45	2	9.72	0.00
c2_10_10	5549.16	168.07	1	5784.16	168.07	1	4.06	0.00
r2_10_1	7766.3	237.64	2	8157.02	237.64	2	4.79	0.00
r2_10_2	12,035.15	380.46	2	12,221.02	375.94	2	1.52	1.20
r2_10_3	7881.84	251.32	2	8607.83	250.81	2	8.43	0.20
r2_10_4	11,418.07	349.08	2	12,341.32	349.08	2	7.48	0.00
r2_10_5	12,255.84	388.45	2	12,480.01	372.84	2	1.80	4.19
r2_10_6	8061.82	245.7	2	8790.91	245.7	2	8.29	0.00
r2_10_7	10,146.98	309.57	2	10,387.47	309.57	2	2.32	0.00
r2_10_8	9891.95	311.9	2	10,564.24	311.4	2	6.36	0.16
r2_10_9	10,933.1	346.47	2	11,621.66	346.47	2	5.92	0.00
r2_10_10	8390.58	264.89	2	8390.58	264.89	2	0.00	0.00
rc2_10_1	12,378.44	366.93	2	12,510.12	366.76	2	1.05	0.05
rc2_10_2	15,168.42	447.75	2	15,407.31	444.82	2	1.55	0.66
rc2_10_3	11,685.37	348.53	2	12,362.64	348.53	2	5.48	0.00
rc2_10_4	6313.89	200.49	1	6492.82	198.76	1	2.76	0.87
rc2_10_5	16,706.03	500.43	3	16,939.97	500.43	3	1.38	0.00
rc2_10_6	13,039.36	389.68	2	13,237.28	389.68	2	1.50	0.00
rc2_10_7	12,077.17	355.23	2	12,729.77	355.23	2	5.13	0.00
rc2_10_8	14,736.49	445.55	2	15,187.89	445.55	2	2.97	0.00
rc2_10_9	16,639.49	488.02	2	16,818.09	488.02	2	1.06	0.00
rc2_10_10	9768.34	289.48	2	10,298.78	289.48	2	5.15	0.00

In order to determine the solution quality of the proposed methodology, the relative error (RE) is obtained as follows:

$$RE = \frac{(Z_{PA} - UB)}{UB} \times 100 \qquad (14)$$

where Z_{PA} is the objective value found by the proposed algorithm while UB is the upper bound found by the mathematical programming model obtained by CPLEX which is one of the most powerful commercial solvers. CPLEX solves the problem with an exact algorithm to find the optimum solution. If an optimum solution is found within 3 h, then UB is equal to the optimum value of the corresponding instance.

The proposed algorithm was coded in Microsoft Visual C++ Version 10.0. The computational results obtained for the instances with ten customers are given at Table 3. The name of the instances are placed at the first column. The second column up to the forth represent the optimum fuel consumption $\left(Z_{opt}\right)$, computational time which refers to CPU time given by seconds and the number of vehicles (#vhc) obtained by CPLEX, respectively. The minimum and maximum fuel consumption values obtained from 100 runs of proposed algorithm are given at fifth and sixth column, respectively. The seventh column stands for the total CPU times of 100 runs in terms of seconds. The eighth column gives the number of vehicles found at the run in which the minimum fuel consumption is received. Finally, the number of runs obtaining the optimum value in 100 runs

Table 3 Results for instances with 10 customers

	CPLEX			Proposed Algorithm			
Instance	Z_{opt}	CPU	# vhc	Z_{PA}	CPU	#vhc	#opt
c2_10_1	6643.99	2.043	1	6643.99	0.858	1	9
c2_10_2	5486.09	17.744	1	5486.09	0.837	1	73
c2_10_3	13,542.74	61.769	2	13,542.74	1.048	2	96
c2_10_4	10,348.78	59.873	2	10,348.78	0.979	2	4
c2_10_5	6494.84	8.483	1	6494.84	0.958	1	79
c2_10_6	11,396.71	186.904	2	11,396.71	0.988	2	25
c2_10_7	12,606.8	105.378	2	12,606.8	1.118	2	7
c2_10_8	10,279.76	71.486	2	10,279.76	0.917	2	76
c2_10_9	7406.08	2.436	2	7406.08	0.906	2	96
c2_10_10	5549.16	3.778	1	5549.16	0.897	1	94
r2_10_1	7766.3	54.538	2	7766.3	0.849	2	67
r2_10_2	12,035.15	293.028	2	12,035.15	1.071	2	96
r2_10_3	7881.84	21.039	2	7881.84	0.961	2	50
r2_10_4	11,418.07	14.04	2	11,418.07	0.946	2	94
r2_10_5	12,255.84	17.114	2	12,323.72	0.996	2	0
r2_10_6	8061.82	54.43	2	8061.82	0.829	2	65
r2_10_7	10,146.98	187.799	2	10,146.98	1.107	2	7
r2_10_8	9891.95	79.266	2	9891.95	1.044	2	71
r2_10_9	10,933.1	36.44	2	10,933.1	0.909	2	92
r2_10_10	8390.58	11.68	2	8390.58	0.878	2	77
rc2_10_1	12,378.44	8.001	2	12,378.44	1.03	2	98
rc2_10_2	15,168.42	569.578	2	15,168.42	1.058	2	25
rc2_10_3	11,685.37	188.142	2	11,685.37	0.899	2	89
rc2_10_4	6313.89	11.49	1	6313.89	0.899	1	49
rc2_10_5	16,706.03	2176.548	3	16,706.03	1.073	3	95
rc2_10_6	13,039.36	7.99	2	13,039.36	1.014	2	66
rc2_10_7	12,077.17	8.554	2	12,077.17	1.009	2	9
rc2_10_8	14,736.49	19.1	2	14,736.49	0.93	2	40
rc2_10_9	16,639.49	38.032	2	16,639.49	1.216	2	99
rc2_10_10	9768.34	16.799	2	9768.34	0.95	2	100

Table 4 Results for instances with 20 customers

Instance	CPLEX UB	CPLEX # vhc	Proposed Algorithm Z_{PA}	Proposed Algorithm CPU	Proposed Algorithm #vhc	RE
c2_20_1	17,340.64	4	17,367.13	1.006	3	0.15
c2_20_2	16,684.83	4	16,962.32	0.987	3	1.66
c2_20_3	21,156.52	4	**21,117.02**	1.066	4	−0.19
c2_20_4	20,816.05	4	**20,643.1**	1.039	3	−0.83
c2_20_5	11,890.8	3	11,937.31	1.024	2	0.39
c2_20_6	16,778.96	4	**16,632.65**	0.929	3	−0.87
c2_20_7	22,579	4	22,690.27	1.058	4	0.49
c2_20_8	17,504.05	4	17,504.05	1.034	3	0
c2_20_9	17,622.83	4	17,860.08	1.01	3	1.35
c2_20_10	15,299.7	4	**15,291.04**	0.929	3	−0.06
r2_20_1	13,678.98	4	13,734.11	0.939	3	0.4
r2_20_2	18,584.09	4	18,584.1	0.902	3	0
r2_20_3	13,316.2	4	13,327.53	0.997	3	0.09
r2_20_4	19,071.53	4	19,463.56	0.974	3	2.06
r2_20_5	18,313.13	4	**18,234.01**	1.019	3	−0.43
r2_20_6	15,289.17	4	15,778.01	0.976	3	3.2
r2_20_7	15,992.16	4	**15,962.56**	0.887	3	−0.19
r2_20_8	19,793.08	4	20,318.84	1.081	3	2.66
r2_20_9	19,714.3	4	19,885.11	1.07	3	0.87
r2_20_10	15,029.73	4	15,029.73	0.976	3	0
rc2_20_1	25,816.35	4	26,173.38	1.179	4	1.38
rc2_20_2	28,639.15	5	28,639.14	1.189	4	0
rc2_20_3	20,247.42	4	**20,080.34**	1.138	3	−0.83
rc2_20_4	17,618.93	4	**17,602.8**	0.966	3	−0.09
rc2_20_5	25,145.39	5	**24,520.91**	1.003	4	−2.48
rc2_20_6	23,704.11	4	**23,393.69**	1.103	4	−1.31
rc2_20_7	17,181.58	4	17,181.59	1.102	3	0
rc2_20_8	24,655.92	4	24,841.44	0.942	4	0.75
rc2_20_9	26,764.81	5	26,764.81	1.074	4	0
rc2_20_10	25,718.16	4	26,073.15	1.06	4	1.38

is given at the last column (#opt). According to the computational results, the proposed algorithm obtained optimal solution for all instances except one (r2_10_5), which has a relative deviation of 0.55 %. In comparison with CPLEX, good solutions are acquired in a very short computational time (less than 1.3 s) with the proposed algorithm for the instances with ten customers. The proposed algorithm found the optimal solution in 61 runs out of 100, on average.

The results for the instances with 20 customers are given at Table 4. Since exact solutions cannot be obtained in a reasonable time, the best integer solution found within 3 h is considered as upper bound (UB) for each instance. The RE values are seen at the last column. The proposed algorithm

Table 5 Reduction of fuel consumption for big instances

Instance	N	Fuel consumption O_1	Fuel consumption O_2	CPU	Reduction (%)
C1	50	49,653.9	43,976.8	0.24	11.43
C2	75	75,611.6	72,937.0	0.58	3.54
C3	100	93,789.3	80,722.1	0.66	13.93
C4	150	112,874.1	112,084.8	1.04	0.70
C5	199	146,555.8	145,052.3	1.65	1.03
C6	50	49,653.9	45,015.8	0.23	9.34
C7	75	75,611.6	71,826.9	0.54	5.01
C8	100	94,945.7	80,722.1	0.54	14.98
C9	150	112,874.1	112,084.8	1.03	0.70
C10	200	146,555.8	145,052.3	1.60	1.03
C11	120	141,060.4	131,085.2	1.77	7.07
C12	100	96,244.22	94,379.53	0.54	1.94
C13	120	141,060.4	132,336.2	1.76	6.18
C14	100	96,244.2	93,598.2	0.52	2.75
T1	75	1,120,041	995,237.3	0.97	11.14
T2	75	1,031,607	857,386.5	0.93	16.89
T3	75	802,670.4	562,608.1	1.09	29.91
T4	75	951,940.8	892,198	0.64	6.28
T5	100	1,260,374	1,105,290	2.06	12.30
T6	100	1,622,756	1,502,535	2.22	7.41
T7	100	1,281,038	1,125,065	1.15	12.18
T8	100	1,002,554	860,994.8	1.65	14.12
T9	150	1,866,386	1,706,999	3.87	8.54
T10	150	2,500,952	2,051,358	6.80	17.98
T11	150	1,862,943	1,521,058	1.80	18.35
T12	150	2,282,970	2,081,800	4.60	8.81
T13	385	279,428.7	274,630.2	27.93	1.72
G1	240	512,250.1	497,024	13.04	2.97
G2	320	686,311.8	668,009.7	29.55	2.67
G3	400	815,390.3	805,770.6	191.67	1.18
G4	480	962,291.9	949,609.9	103.23	1.32
G5	200	444,257.2	414,256.7	7.24	6.75
G6	280	604,399.2	582,050.9	21.67	3.70
G7	360	767,978.1	712,448.6	79.02	7.23
G8	440	883,591.6	872,358.8	123.38	1.27
G9	255	302,502.9	290,333.2	6.84	4.02
G10	323	373,940.8	367,464.1	13.53	1.73
G11	399	454,631.3	450,158.4	29.89	0.98
G12	483	535,604	530,766.3	40.24	0.90
G13	252	446,474.1	437,474.7	3.60	2.02
G14	320	575,984.9	563,333.1	9.14	2.20
G15	396	714,551.9	709,811	18.16	0.66
G16	480	854,464.2	848,425.7	37.26	0.71
G17	240	85,821.7	83,028.59	3.00	3.25
G18	301	118,057.9	118,012	4.53	0.04
G19	360	168,759.7	166,054.6	13.06	1.60
G20	420	220,956.8	219,917.1	13.04	0.47

acquired fuel consumption less than UB for 10 instances, which are written in bold in Table 4. Although the UB is not obtained for 14 instances, the biggest RE value is 2.66 %, i.e. the results of the proposed algorithm are only 2.66 % far away from the UB. Moreover, the solutions found by the proposed algorithm require less vehicles than UB for most of the instances. All results are found in at most 1.2 s while these UB values are obtained by CPLEX in 3 h.

Computational experiments are also performed for well-known big size instances from the literature of VRP in order to investigate the reduction of fuel consumption obtained by the proposed algorithm. Since exact algorithms from the literature cannot solve MTVRP instances having more than 50 customers in an appropriate computational time [18], the results of proposed algorithms cannot be compared with the optimum result of MIP. Data sets generated by Christofides et al. [12], Taillard [41] and Golden et al. [17] for capacitated VRP are used as benchmark instances. There are 14 instances in the data set of Christofides et al. [12] (C1–C14) in which the number of customers changes between 50 and 200. The data set of Taillard [41] (T1–T13) includes 13 instances between 75 and 385 customers. Finally, there are 20 instances in the data set of Golden et al. [17] (G1–G20) ranging from 240 to 483. The parameters are as the ones used for Solomon's instances. Computational results are given in Table 5. The name of the instances are given at the first column. The second column represents the number of the customers. The algorithm runs ten times for all instances to minimize distance, and the minimum fuel consumption values obtained from distance minimization are given at the third column. Then the algorithm runs to minimize fuel consumption and the minimum fuel consumption values are given at the fourth column. O_1 represent the fuel consumption value obtained by the distance minimization while O_2 is obtained by the fuel consumption minimization. The fuel consumption values given at the third and forth columns are obtained from ten runs for each instances for each objective function (distance minimization and fuel consumption minimization). The fifth column represents the total CPU time of ten runs for each instance. The last column gives the obtained fuel consumption reduction which is given as the following:

$$reduction = \frac{O_1 - O_2}{O_1} \times 100 \tag{15}$$

According to the results given in Table 5, on average, 9.27 % fuel consumption reduction can be obtained for instances having 50–200 customers (C1–C14, T1–T12 and G5) in 1.77 s. Up to 30 % reduction can be obtained for these instances. The average fuel reduction for the instances having size 240–483 (T13, G1–G4, G6–G20) is 2.03 %. Although the fuel consumption reductions are not as much as smaller instances, the computational time is still less then 3 min for big instances.

7 Concluding Remarks

In this study, a MIP model is developed and a solution methodology based on SA is proposed for CumMTVRP-LD. It is shown that up to 10 % reduction in fuel consumption can be achieved without substantial increase in distance and the number of vehicles for instances with ten customers. Moreover, the proposed SA-based methodology found promising results for CumMTVRP-LD minimizing total fuel consumption for large size instances.

Initial solution quality is very important for SA. The saving algorithm used in this work generates the same initial solution in every run of an instance. This may cause lack of solution diversity by getting trapped in a local optimum during SA. In further studies, the saving algorithm will be improved to generate various good quality solutions. Finally, the operators generating initial solutions can be improved for CumMTVRP-LD. The methodology will also be enhanced for CumMTVRP-LD with a heterogeneous fleet.

In this study, the CumMTVRP-LD is modeled deterministically. Because of the several unexpected factors, such as road and weather conditions, most of the real-life routing problems contain elements of uncertainty [20]. The CumMTVRP-LD investigated in this study can be studied with various stochastic parameters in further studies.

Acknowledgments This research is partially supported by LATNA Laboratory, NRU HSE, RF government grant, ag. 11.G34.31.0057., and D. Cinar is supported by TUBITAK- BIDEB 2219-International Postdoctoral Research Scholarship Program during this study at the University of Florida. We thank the anonymous reviewers for providing insightful comments and directions which has resulted in this paper.

References

1. Alizamir, S., Rebennack, S., Pardalos, P.M. (2008). Improving the neighborhood selection strategy in simulated annealing using the optimal stopping problem. In C.M. Tan (Ed.) *Simulated annealing*, (pp. 363–382). InTech.
2. Alonso, F., Alvarez, M.J., Beasley, J.E. (2008). A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society*, *59*(7), 963–976.
3. Andress, D., Nguyen, T.D., Das, S. (2011). Reducing GHG emissions in the United States' transportation sector. *Energy for Sustainable Development*, *15*(2), 117–136.
4. Azi, N., Gendreau, M., Potvin, J.Y. (2007). An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research*, *178*(3), 755–766.

5. Azi, N., Gendreau, M., Potvin, J.Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, *202*(3), 756–763.
6. Azi, N., Gendreau, M., Potvin, J.Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, *41*, 167–173.
7. Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, *8*(1), 10–15.
8. Brandão, J., & Mercer, A. (1997). A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, *100*(1), 180–191.
9. Brandão, J., & Mercer, A. (1998). The multi-trip vehicle routing problem. *Journal of the Operational Research Society*, *49*(8), 799–805.
10. Breedam, A.V. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research*, *86*(3), 480–490.
11. Cattaruzza, D., Absi, N., Feillet, D., Vidal, T. (2013). A memetic algorithm for the multi trip vehicle routing Problem. *European Journal of Operational Research*. doi:10.1016/j.ejor.2013.06.012.
12. Christofides, N., Mingozzi, A., Toth, P. (1979). Combinatorial Optimization.
13. Clarke, G., & Wright, J.W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, *12*(4), 568–581.
14. Şen, A., & Bülbül, K. (2008). A survey on multi trip vehicle routing problem. In *Proceedings of the 6th International Logistics and Supply Chain Congress,* 6–7 November, (pp. 401–407). Istanbul, Turkey.
15. Taillard, É.D., Laporte, G., Gendreau, M. (1996). Vehicle routeing with multiple use of vehicles. *The Journal of the Operational Research Society*, *47*(8), 1065–1070.
16. Fleischmann, B. (1990). The vehicle routing problem with multiple use of vehicles. *Working paper*. Universitt Hamburg, Germany: Fachbereich Wirtschaftswissenschaften.
17. Golden, B., Wasil, E., Kelly, J., Chao, I. (1998). Metaheuristics in vehicle routing. In T.G. Crainic, & G. Laporte (Eds.) *Fleet Management and Logistics*, (pp. 49–58). Boston: Kluwer Academic Publishers.
18. Hernandez, F., Feillet, D., Giroudeau, R., Naud, O. (2014). A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4OR*, *12*(3), 235–259. doi:10.1007/s10288-013-0238-z.
19. Huang, C., & ve Lee, C. (2011). A study of multi-trip vehicle routing problem and distribution center location problem. In *Proceedings of POMS 22nd Annual Conference*. Reno, Nevada, USA.
20. Huang, Z., & Zheng, Q. (2014). Decomposition-based exact algorithms for risk-constrained traveling salesman problems with discrete random arc costs, (pp. 1–16).
21. International Energy Agency (2011). United States: Indicators for 2011. http://www.iea.org/statistics/statisticssearch/.
22. Kara, İ., Kara, B.Y., Yetiş, M.K. (2008). Cumulative vehicle routing problems. In T. Caric, & H. Gold (Eds.), *Vehicle Routing Problem* (pp. 85–98). KG, Vienna: I-Tech Education and Publishing.
23. Kokubugata, H., & Kawashima, H. (2008). In C.M. Tan (Ed.) *Application of simulated annealing to routing problems in city logistics*, (pp. 131–154). InTech.
24. Kopfer, H., & Kopfer, H. (2013). In H.J. Kreowski, B. Scholz-Reiter, K.D. Thoben (Eds.) *Emissions minimization vehicle routing problem in dependence of different vehicle classes*, (pp. 49–58). Berlin Heidelberg: Springer.
25. Lenstra, J.K., & Kan, A.H.G.R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, *11*(2), 221–227.
26. Li, J. (2012). Vehicle routing problem with time windows for reducing fuel consumption. *Journal of Computers*, *7*(2), 3020–3027.
27. Lin, C., Choy, K., Ho, G., Chung, S., Lam, H. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, *41*, 1118–1138.
28. Ma, X., Huang, Z., Koutsopoulos, H. (2014). Integrated traffic and emission simulation: a model calibration approach using aggregate information. *Environmental Modeling & Assessment*. doi:10.1007/s10666-013-9397-8.
29. Ma, X., Lei, W., Andrasson, I., Chen, H. (2012). An evaluation of microscopic emission models for traffic pollution simulation using on-board measurement. *Environmental Modeling & Assessment*, *17*(4), 375–387.
30. Macedo, R., Alves, C., de Carvalho, J.V., Clautiaux, F., Hanafi, S. (2011). Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, *214*(3), 536–545.
31. Marinakis, Y., Migdalas, A., Pardalos, P. (2007). A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. *Journal of Global Optimization*, *38*(4), 555–580.
32. Mingozzi, A., Roberti, R., Toth, P. (2013). An exact algorithm for the multitrip vehicle routing problem. *INFORMS Journal on Computing*, *25*(2), 193–207.
33. Nanduri, V., & Kazemzadeh, N. (2012). A survey of carbon market mechanisms and models. In Q.P. Zheng, S. Rebennack, P.M. Pardalos, M.V.F. Pereira, N.A. Iliadis (Eds.) *Handbook of CO_2 in Power Systems, Energy Systems*, (pp. 89–105). Berlin Heidelberg: Springer.
34. Nguyen, P.K., Crainic, T.G., Toulouse, M. (2013). A tabu search for time-dependent multi-zone multi-trip vehicle routing Problem with Time Windows. *European Journal of Operational Research*, *231*(1), 43–56.
35. Olivera, A., & Viera, O. (2007). Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research*, *34*(1), 28–47.
36. Petch, R., & Salhi, S. (2004). A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics*, *133*(1–3), 69–92.
37. Report to Congress U.S. Department of Transportation (2010). *Transportation'srole in reducing U.S. greenhouse gas emissions*. Volume 1: Synthesis Report, Department of Transportation, USA.
38. Salhi, S., & Petch, R.J. (2007). A GA based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modelling and Algorithms*, *6*(4), 591–613.
39. Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265.
40. Svirezhev, Y., Brovkin, V., von Bloh, W., Schellnhuber, H., Petschel-Held, G. (1999). Optimisation of reduction of global CO_2 emission based on a simple model of the carbon cycle. *Environmental Modeling & Assessment*, *4*(1), 23–33.
41. Taillard, E. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, *23*(8), 661–673.
42. Tavakkoli-Moghaddam, R., Gazanfari, M., Alinaghian, M., Salamatbakhsh, A., Norouzi, N. (2011). A new mathematical model for a competitive vehicle routing problem with time windows solved by simulated annealing. *Journal of Manufacturing Systems*, *30*(2), 83–92.
43. Yellow, P.C. (1970). A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly*, *21*(2), 281–283.