

A modified clustering search based genetic algorithm for the proactive electric vehicle routing problem

Issam El Hammouti^{a*}, Khaoula Derqaoui^a and Mohamed El Merouani^a

^aDepartment of Mathematics, Faculty of Science of Tetuan, Morocco

CHRONICLE

Article history:

Received March 24 2023

Received in Revised Format

July 10 2023

Accepted September 11 2023

Available online

September 11 2023

Keywords:

Meta-heuristics

Mathematical modelling

Clustering

Genetic algorithm

Electric vehicle routing

Travel time uncertainty

ABSTRACT

In this paper, an electric vehicle routing problem with time windows and under travel time uncertainty (U-EVRW) is addressed. The U-EVRW aims to find the optimal proactive routing plan of the electric vehicles under the travel time uncertainty during the route of the vehicles which is rarely studied in the literature. Furthermore, customer time windows, limited loading capacities and limited battery capacities constraints are also incorporated. A new mixed integer programming (MIP) model is formulated for the proposed U-EVRW. In addition to the commercial CPLEX Optimizer version 20.1.0, a modified Clustering Search based Genetic algorithm (MCSGA) is developed as a solution method. Numerical tests are conducted on the one hand to validate the effectiveness of the proposed MCSGA and on the other hand to analyze the impact of travel time uncertainty of the electric vehicle on the solutions quality.

© 2023 by the authors; licensee Growing Science, Canada

1. Introduction

Nowadays, to manage climate change and reduce greenhouse gases emission, the transportation system is forced to operate with the safest fuel sources instead of the usage of petroleum-based fuel. One of the cleanest and most achievable fuel sources is electricity. Indeed, the most important delivery companies have incorporated electric vehicles (EVs) into their fleet. Operating with EVs have many advantages, for example, lower CO₂ emissions, less maintenance and minimal noise. However, due to the limited battery power, these types of vehicles cannot travel long distances without visiting a charging station, which is time and cost consuming. Therefore, the first concern of delivery companies is to find the optimal routing of EVs which minimizes the time and the cost of deliveries. This problem is a variant of the classical vehicle routing problem (VRP) and is commonly known as the electric vehicle routing problem (EVRP). The EVRP consists in using electric vehicles to complete the last mile delivery operations with a minimum cost route (Toth et Vigo., 2014). For large details on EVRP, please refer to (Qin et al., 2021). Most papers on EVRP addressed the problem in a deterministic environment. However, in the real-world case, unexpected events are usually present. For example, the travel time between two points depends on several factors such as the state of route, the weather, the traffic jump, or the possible accidents. These unexpected factors can increase the estimated travel time. Consequently, a drastic change in routing plans should be applied at the last moment. Hence, planners should anticipate these unexpected events to avoid disruptions in their initial planning. This work addresses how to find a routing plan of the electric vehicle under travel time uncertainty. A proactive strategy that considers the occurrence probability of a finite set of discrete scenarios that can happen during the route planning is proposed. The contributions of this paper are a novel mixed integer mathematical modeling for the U-EVRW and a modified clustering search based genetic algorithm with a Multi-Parent order Crossover operator (MCSGA) to solve it. The rest of the paper is structured as follows: The next section presents a literature review on the EVRP. Then, a mathematical model for U-EVRW is presented in Section 3. In Section 4,

* Corresponding author

E-mail: issam.elhammouti@gmail.com (I. E. Hammouti)

ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)

2023 Growing Science Ltd.

doi: 10.5267/j.ijiec.2023.9.004

a detailed description of the proposed MCSGA is presented. Next, computational experiments and comparison results are exposed in Section 5. Finally, Section 6 concludes the paper.

2. Related works

Papers dedicated to logistics activities are increasingly a fertile area of operations research, with the vehicle routing problem (VRP) being one of the most studied and well-known transport problems that consider the increasing use of EVs (Afroditi et al., 2014). In general, studies on EVRP aim to provide decision support tools and to encourage the switch to green energy, including electric vehicles. Over the years, many interesting extensions of VRP have been proposed by including different real-world constraints. First, (Erdoğan & Miller-Hooks, 2012) proposed a mixed integer linear program for the Green Vehicle Routing Problem (GVRP) and solved this model by a modified Clarke and Wright savings heuristic and a Density-Based Clustering Algorithm. Then, Schneider et al. (2014) presented the study of the Electric Vehicle Routing Problem with Time Windows (E-VRPTW). They developed a hybrid heuristic that combines a variable neighborhood search algorithm with a tabu search heuristic to solve the problem. Later, the EVRP is studied by different authors. Nowadays EVRP is one of the most important extensions of VRP, due to its environmental benefits. This literature review will progressively highlight articles on VRP published since the year 2014 that reviewed aspects of green logistics and new solution approaches. (Allahviranloo et al., 2014) proposed three new formulations to address the selective vehicle problem under uncertain demand level. Three Parallel genetic algorithms are developed to solve these formulations at large scale level. The results proposed by the authors show that the Parallel genetic algorithm wherein communication between subpopulations, does not eliminate repeated chromosomes, outperforms other algorithms at level of computation time.

The authors in (Keskin & Çatay, 2016) opted for partial charging restriction instead of full charging; they proposed a binary mixed integer linear program and developed an adaptive large neighborhood search (ALNS) algorithm to solve it. In (Hiermann et al., 2019), the authors proposed a mixed integer programming model for the EVRP. They considered a fleet with different types of electric vehicles. For small instances, the authors proposed a branch-and-price algorithm as a solution method, while for large instances; they developed a hybrid ALNS-based heuristic with integrated local search. The work given in (Montoya et al., 2017) is the first which addressed the EVRP with a nonlinear charging function for the battery-charge level. According to the results the authors suggest that neglecting nonlinear charging may lead to infeasible or overly expensive solutions. (Cao et al., 2018) proposed a hybrid algorithm based on an artificial immune system for VRP with interval requests. The authors in (Zhang & Tang, 2009) proposed a hybridization of the ant colony optimization (ACO) with scatter search (SS) as a new metaheuristic solution to solve the VRP. In (Verma, 2018), the authors presented a new variant of the E-VRPTW by allowing the available stations to serve both as a traditional recharging station and as a battery swapping station. Hiermann et al. (2016) addressed the VRP with consideration of three types of vehicles, conventional, Plug-in hybrid, and electric vehicles. They developed a metaheuristic which combines a genetic algorithm with local and large neighborhood search to solve this problem. In (Macrina et al., 2019), the authors developed an iterative local search heuristic to solve the VRP with both electric and internal combustion engine vehicles. To respect environmental measures, limitations on the polluting emissions for the internal combustion engine vehicles are considered. (Cortés-Murcia et al., 2019) proposed a new variant of the electric vehicle routing problem in which an alternative vehicle (bikes, drones, segways, etc.) is used to customer visits while the electric vehicle is at recharging station. In (Koç et al., 2019) the authors address an extended version of the electric vehicle routing problem with nonlinear charging function by considering several companies that shared charging stations to determine the routes of their vehicle fleet. The objective of this paper is to find the best location and technology of the charging stations to minimize the cost of travel time of the vehicles of each company. (Pelletier et al., 2019) propose a robust mixed integer linear program for the electric vehicle routing problem under uncertainties in energy consumption due to several factors such as the weather, road conditions and driver behavior. They developed a two-phase heuristic method based on Large Neighborhood Search to solve larger instances of the problem. In (Zhao & Lu, 2019), the authors proposed a heuristic approach based on the Adaptive Large Neighborhood Search (ALNS) and integer programming to solve VRP taken into account customer time windows, limited loading capacities and limited battery capacities of the vehicles. Wang et al (Wang et al., 2019) extended the multi-depot VRPTW problem by considering shared transport resources to minimize the total carbon emission as well as the operational cost. They developed a hybrid metaheuristic which combines the Clarke and Wright Savings Heuristic Algorithm (CWSHA), the Sweep Algorithm (SwA) and the Multi-Objective Particle Swarm Optimization algorithm as a method of resolution. In (Mao et al., 2020), the authors developed a hybrid algorithm which combine An Improved Ant Colony Optimization Algorithm and a Local Search to solve the EVRP with multiple recharging options, which are partial recharging and battery swapping. (Messaoud, 2021) proposed an extension of the EVRP by considering the Stochastic Travel Times (EVRPSTT). To solve this problem, the author developed a Chance Constrained Programming (CCP) Model, as well as an Improved Large Neighborhood Search (ILS) algorithm and a Monte Carlo Sampling (MCS) procedure. In (Pan & Liu, 2023), the authors proposed a formulation based on a deep reinforcement learning framework to deal with the dynamic characteristics of the route in real-world urban logistics. A deep neural network with a dynamic attention mechanism is used to collect the data on the changes in customers' demands in real-time and a cutting-edge reinforcement learning algorithm is presented for better training the routing process's dynamics and uncertainty.

3. Mathematical model

This section describes an extension of the mathematical formulation presented in (Schneider et al., 2014) for the E-VRPWT. The proposed model incorporates a proactive strategy which considers the travel time uncertainty of the vehicles. In E-VRPWT a fleet of EVs with fixed loading capacities and limited battery power are used for the package delivery from a single depot to a set of customers. Customers are characterized by their demands, delivery time windows and service durations. To complete the delivery operation, the EVs may need to visit a station to recharge the consumed battery during the route. In (Schneider et al., 2014) model's, the authors considered a deterministic travel time between nodes. However, in real world case, due to many unforeseen factors that can affect the travel time of vehicles between nodes, that quantity is highly uncertain and may have a significant impact on the initial planning. In this work, we propose a proactive model that takes into account the travel time uncertainty of vehicles by the integration of the occurrence probability of unexpected scenarios. To clarify the proposed mathematical model, the following notations are considered:

Assumption

The assumptions represent the real-world applications:

- Each customer will be served at most once.
- Each customer will be only visited by one vehicle.
- A station can be visited more than once.
- The total demands of customers visited by one vehicle cannot exceed its load capacity.
- Any vehicle can serve any customer.
- The demands, locations, and time windows of customers are known in advance.
- Routes begin and end at the depot.
- The vehicle speed is assumed the same for all vehicles and for simplification and to have distance=time the speed is assumed equal to 1.
- The vehicle's traveling time is considered uncertain.

Sets

V : Set of customers vertices.

$i = (0, N+1)$ represents the same depot node, and every route starts at the depot 0 and ends at the depot $N + 1$.

F : Set of recharging stations

F' : Set of dummy vertices generated to enable several visits to each vertex in the set F .

F'_0 : Set of recharging stations including the starting depot 0.

F'_{N+1} : Set of recharging stations including the ending depot $N+1$.

V_0 : Set of customers including the starting depot 0.

V' : Set of customer vertices and recharging stations, $V' = V \cup F$.

V'_0 : Set of customers and recharging stations including the starting depot 0 ($V'_0 = V' \cup 0$).

V'_{N+1} : Set of customers and recharging stations including the ending depot $N+1$ ($V'_{N+1} = V' \cup N + 1$).

$V'_{0,N+1}$: Set of customers and recharging stations including starting and ending depot ($V'_{0,N+1} = V' \cup \{0\} \cup \{N + 1\}$).

W : Set of scenarios.

Parameters

LC = Loading capacity of EVs

Q = Battery capacity of EVs

$CostV$ = Company usage cost of one vehicle (is assumed equal to 50\$)

$CostT$ = Cost of travel time per minute (is assumed equal to 0.15 \$)

g = Recharging rate

h = Charge consumption rate

s_i = Service time at node i , 0 if $i \in V$

e_i = Earliest start of service at node $i \in V'_{0,N+1}$

l_i = Earliest start of service at node $i \in V'_{0,N+1}$

q_i = Demand of node i , ($q_i = 0$ if $i \in F$)

P_k = Occurrence probability of each scenario $k \in w$

d_{ij} = Distance between nodes i and j (we note that d_{ij} is the Euclidean distance between nodes i and j)

t_{ij} = Estimated travel time between nodes i and j

ts_{ij}^k = Travel time between nodes i and j in scenario $k \in w$ (uncertain travel time)

Decision Variables

x_{ij} : is equal 1 if the route between i and j is traversed 0 otherwise,
 G_i : Service start time of node i ,
 G_s^k : Service start time of node i in scenario k ,
 U_i : Remaining cargo on arrival at vertex i ,
 U_s^k : Remaining cargo on arrival at vertex i in scenario k ,
 bt_i : Remaining charge battery on arrival at vertex i ,
 bts_i^k : Remaining charge battery on arrival at vertex i in scenario k

Objective function

$$\min costT * (\sum_{i \in V'_0} \sum_{j \in V'_{N+1} / i \neq j} t_{ij} \times x_{ij} + \sum_{k \in w} \sum_{i \in V'_0} \sum_{j \in V'_{N+1} / i \neq j} P_k \times (ts_{ij}^k - t_{ij}) \times x_{ij}) + \sum_{j \in V'} x_{0j} \times costV \quad (1)$$

subject to

$$\sum_{j \in V'_{N+1} / i \neq j} x_{ij} = 1, \quad \forall i \in V \quad (2)$$

$$\sum_{j \in V'_{N+1} / i \neq j} x_{ij} \leq 1, \quad \forall i \in F \quad (3)$$

$$\sum_{j \in V'_{N+1} / i \neq j} x_{ij} - \sum_{j \in V'_0 / i \neq j} x_{ji} = 0, \quad \forall i \in V' \quad (4)$$

$$G_i + (s_i + t_{ij}) * x_{ij} - l_0 * (1 - x_{ij}) \leq G_j, \quad \forall i \in \{0\} \cup V, \forall j \in V'_{N+1} / j \neq i \quad (5)$$

$$G_i + ts_{ij}^k * x_{ij} + G_s^k - l_0 * (1 - x_{ij}) \leq G_j + G_s^k, \quad \forall k \in w, \forall i \in \{0\} \cup V, \forall j \in V'_{N+1} / j \neq i, \quad (6)$$

$$G_i + t_{ij} * x_{ij} + (g * (Q - bt_i)) - (l_0 + g * Q) * (1 - x_{ij}) \leq G_j, \quad \forall i \in F', \forall j \in V'_{N+1} / j \neq i \quad (7)$$

$$G_i + ts_{ij}^k * x_{ij} + G_s^k - (l_0 + g * Q) * (1 - x_{ij}) \leq G_j + G_s^k \quad \forall k \in w, \forall i \in F', \forall j \in V'_{N+1} / j \neq i \quad (8)$$

$$e_j \leq G_i + G_s^k \leq l_j, \quad \forall k \in w, j \in V'_{0,N+1} \quad (9)$$

$$0 \leq U_j \leq U_i - (q_i * (x_{ij}) + LC * (1 - x_{ij})), \quad \forall i \in V'_0, \forall j \in V'_{N+1} / j \neq i \quad (10)$$

$$0 \leq U_0 \leq C \quad (11)$$

$$0 \leq U_s^k \leq U_s^k - (q_i * (x_{ij}) + LC * (1 - x_{ij})), \quad \forall k \in w, \forall i \in V'_0, \forall j \in V'_{N+1} / j \neq i \quad (12)$$

$$0 \leq U_s^k \leq LC, \quad \forall k \in w \quad (13)$$

$$0 \leq bt_j \leq bt_i - (h * t_{ij}) * x_{ij} + Q * (1 - x_{ij}), \quad \forall i \in V, \forall j \in V'_{N+1} / j \neq i \quad (14)$$

$$0 \leq bts_j^k \leq bts_i^k - (h * ts_{ij}^k) * x_{ij} + Q * (1 - x_{ij}), \quad \forall k \in w, \forall i \in V, \forall j \in V'_{N+1} / j \neq i \quad (15)$$

$$0 \leq bt_j \leq Q - (h * t_{ij}) * x_{ij}, \quad \forall i \in F'_0, \forall j \in V'_{N+1} / j \neq i \quad (16)$$

$$0 \leq bts_j^k \leq Q - (h * ts_{ij}^k) * x_{ij}, \quad \forall k \in w, \forall i \in F'_0, \forall j \in V'_{N+1} / j \neq i \quad (17)$$

$$x_{ij} == 0, \quad \forall i, j \in F' \quad (18)$$

$$x_{0j} == 0, \quad \forall j \in F' \quad (19)$$

$$0 \leq x_{ij} \leq 1, \quad \forall i \in V'_0, \forall j \in V'_{N+1} / j \neq i \quad (20)$$

$$G_i, G_s^k, U_i, U_s^k, bt_i, bts_i^k \geq 0 \quad \forall i \in V', \forall k \in w \quad (21)$$

The main objective of the problem is to determine a proactive routing plan to anticipate unexpected scenarios which can alter the expected travel time between nodes. The objective function (1) minimizes both the cost of the total travel time and the cost of the fleet size used in the determined proactive routing plan. Constraints (2) and (3) ensure that each customer is visited exactly once and handle the connectivity of visits to recharging stations. Constraint (4) guarantees that if a customer or a station is visited by a vehicle this last should leave after the end of their services. Constraint (5) ensures that if a vehicle visits a customer j after a customer i , then the service start time of customer j will be greater than or equal to the service start time of customer i plus the service time spent with customer i plus the travel time between customer i and j . Constraint (6) does the same as Constraint (5) but in unexpected scenarios. Constraint (7) ensures that if a vehicle visits a customer j after a recharging station i , then the service start time of customer j will be greater than or equal to the service start time at recharging

station i plus the recharge times for a complete recharge in station i plus the travelled time between i and j . Constraint (8) does the same as Constraint (7) but in unexpected scenarios. Constraint (9) guarantees that each customer is served in its time windows. Constraints (10-13) ensure that each vehicle does not exceed its maximum load capacity. Constraints (14-17) determine the battery level of each vehicle and guarantee that none of them run out of battery. Constraints (18-21) define the type of decision variables (x_{ij} , G_i , G_s^k , U_i , U_s^k , bt_i , bts_i^k).

4. Proposed Modified Clustering Search Based Genetic Algorithm (MCSGA)

Due to the NP-hardness of the EVRP (Schneider et al., 2014), no exact method can solve it for large-sized instances in a reasonable time. For this reason, a Modified Clustering Search Based Genetic Algorithm (MCSGA) is proposed as a new alternative approach to solve it within a short running time. The proposed MCSGA is based on the Clustering Search algorithm (CSA) proposed by (Chaves & Lorena, 2010). In our case the CSA is combined with a genetic algorithm and a new crossover mechanism called the Multi-Parent Order Crossover operator (MPC) to improve exploration capacity of the algorithm. The MCSGA combines three main techniques: a genetic algorithm to generate potential feasible solutions for the problem, a clustering search method to group the solutions into different clusters and find promising regions in the feasible space of the problem, and a local search technique to intensify the search in the promising regions. A cluster (i) is characterized by a center (CT_i), a capacity maximal (CA_{max}) and a number r_i that represents the number of times that a local search is applied consecutively on the cluster i . (r_{max} defines the maximal number of times that a local search can be applied consecutively on the cluster (i) without no improvements in the solution.). (CT_i) represents the position of the cluster (i) in the space search while (CA_{max}) is the number of solutions that can be grouped in any cluster (i). To begin the procedure, the MCSGA generates randomly (m) feasible solutions of the problem. Each generated solution represents the center (CT_i) of the (m) different clusters. Then, a genetic algorithm is applied to generate potential new solutions (section 4.1). Next, a clustering process is executed to allocate the generated new solutions in the most adequate cluster (section 4.2). Finally, a local search technique is applied whenever a cluster is considered as a promising region.

4.1 Genetic algorithm

To generate good solutions for the MCSGA, a genetic algorithm (Holland, 1975) is developed and used as the main components in our CSGA algorithm. Following, the steps of the proposed GA are briefly described.

4.1.1 Solution representation and decoding process

The encoding of the solution is primordial to develop an efficient meta-heuristic. In the proposed algorithm, a solution is represented by an array of $2 \times (N+K+1)$ cells, where N is the number of customers and K is the number of times when the recharging stations are visited. The first and the last cell represent the depot "D" because each vehicle should start and end with the depot, the cells with odd indices may contain an integer number in the interval $[1, N]$. This integer number represents the customer visited by the vehicle or may contains the recharging station "S;" where $i \in [1, M]$ and M is the number of recharging stations. The cells with even indices may contain the depot "D" or the same integer number of the previous cell which means that the vehicle will visit another customer without the need for recharging the power battery. Fig. 1 shows an example of the solution representation:

D	2	2	3	3	S1	D	4	4	1	1	5	D	6	6	7	D
---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	---	---

Fig. 1. Solution representation

To ensure the feasibility of a solution, the following constraints should be respected:

- Each customer should be visited in their windows time.
- A customer is visited by a vehicle if the battery capacity of the vehicle is at least enough to arrive and leave the customer i .
- The loading capacity of each vehicle should be respected.

The decoding of the solution presented in Fig. 1 is the following: three vehicles are used to complete the delivery. The first one starts its route at the depot then the customers 2, 3 and the recharging station S1 are visited consecutively and finally the vehicle returns to the depot "D". The second, begin its route at the depot, then the customers 4, 1 and 5 are visited and finally the vehicle return to the depot "D" and the last one, starts its route at the depot "D", then the customers 6, 7 are visited and finally the vehicle return to the depot "D".

4.1.2 Initial population

The initial population plays a primordial role in fast convergence of genetic algorithms towards the optimal solution because each new population is generated based on the component of the current population. Therefore, to start with potential solutions, the initial population of the used GA is randomly generated except one of them, which is based on the strategy to visit the

maximum customers as possible without visiting a recharging station.

4.1.3 Fitness

In GA, Potential individuals are selected, and weak individuals are eliminated according to their fitness values. The EVRP is a minimization problem; thus, the higher fitness value must be the smaller objective function value.

4.1.4 Selection operator (Tournament process)

At each generation, a tournament process is applied to select the most interesting solutions and update the population. In this process, to give all individuals the chance to be selected, three, four, or five individuals are selected randomly from the current population and the best of them is selected to be part of the new population. The process is repeated until the population is completely updated.

4.1.5 Crossover process

The crossover operation has a primary impact on the efficiency and diversity of the algorithm. Two crossover operators have been applied in the proposed approach, a classic multiple crossover point operator and a multi-parent order crossover operator.

- Multiple Crossover point operator:

This operator consists in mixing the principal components of two solutions (parents) based on multiple crossover points from the current population. The goal of this operator is the generation of two new solutions (offspring) (Fig. 2).

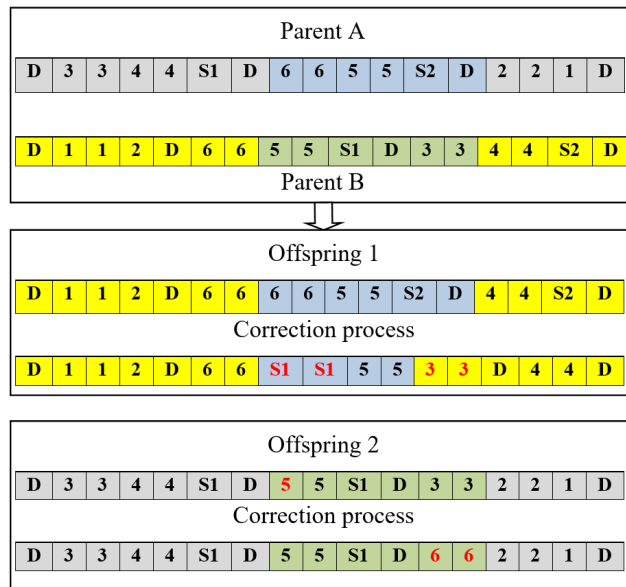


Fig. 2. Multiple Crossover point operator

- Multi-Parent order Crossover operator (MPC):

The MPC consists randomly of a set of solutions of parents instead of two parents and mixed their components to have a single offspring (S_c). The procedure of MPC technique in our case is the followed:

First, the number (cd) of candidate parents to mix is already fixed as an input. After that, we generate randomly (cd-1) crossover points in the interval $[1, 2 \times (N+K+1)]$ where N is the number of customers and K is the number of times when the recharging stations are used. The (cd-1) crossover points divides each parent selected into (cd) parts.

Then, the components of the first part of the first parent are selected as the components of the first part of the offspring solution. Similarly, the components of the second part of the second parent are selected as the components of the second part of the offspring solution and so on until the components of the last part of the last parent which are selected as the components of the last part of the offspring solution (Fig. 3).

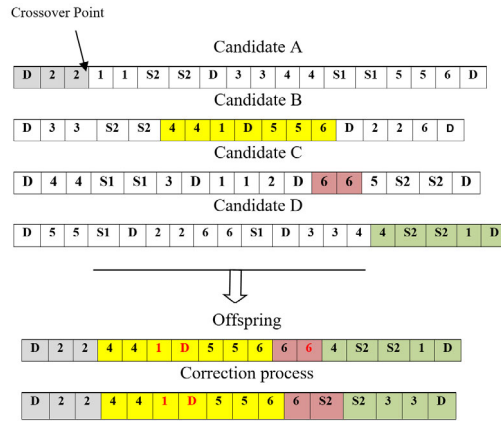


Fig. 3. Multi-Parent order Crossover operator (MPC)

Correction process

After generation of the offspring solution, the following cases can happen:

- ✓ The offspring solution is a legal solution i.e. the offspring respects all constraints of the problem. So, in this case we accept the new solution.
- ✓ The offspring solutions is illegal solution in this case a correction process is applied to ensure the feasibility of the new generated solutions (Fig. 2 and Fig. 3).

An illegal solution is caused by some of the following cases:

- Maybe a customer is visited outside of their windows time, in this case a swapping strategy is applied to place the customer in a feasible window time.
- Maybe a customer is visited by a vehicle, but the battery capacity of this latter does not allow him to reach the nearest station, in this case a swapping strategy is applied between the customer and the nearest charging station.

4.1.6 Mutation

To ensure the diversity of solutions, the MCSGA applies a swap mutation operator on the center CT_i of each promising cluster. Two components from the center CT_i are randomly selected and replaced one by another and vice versa to generate a new solution candidate (Fig. 4).

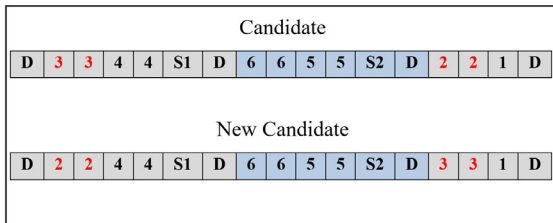


Fig. 4. Mutation Operator

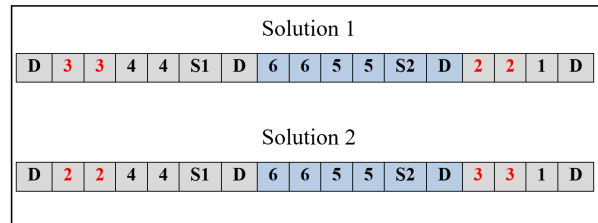


Fig. 5. Example of distance between two solutions

4.2 Clustering process (CP)

The CP consists in comparing each new solution generated by GA to the center CT_i of each cluster i . Then, this new solution is assigned to the cluster with the most similar center. At each generation, the cluster with a new solution updates their center CT_i to change their position within the search space. The similarity between solutions is measured by the distance between them. In this paper, the distance $D(i,j)$ between two solutions i and j is equivalent to the number of cells with different values (Fig. 5). $D(i,j)$ is a positive number that should increase according to the increasing distance between i and j . The purpose of CP is to determine the cluster that represents the most promising region of search. After each iteration of the MCSGA, the number of solutions grouped in each cluster is calculated and if a cluster reaches this maximal capacity (CA_{max}). Thus, this cluster can be considered as a potential region to explore. Then, a local search technique is applied on the center CT_i of each potential cluster i to improve the solution quality. Finally, a mutation operator is applied on the center CT_i of each potential cluster i to avoid possible local optimum. This mutation operator is applied only after the application of the local search technique a maximal number of times (r_{max}) consecutively and no improvement in solution quality is observed. Ending this CP, returns the GA that will generate a new solution. The MCSGA stops after a number maximal (I_{max}) of iterations.

4.3 Local search

After each iteration, a local search strategy is applied to improve the solution quality. This strategy is based on the following two procedures:

- Swap two customers who are in the planned route of a given vehicle. The objective of this swap operation is to change the order in which the customers are visited.
- Swap two customers who are in the planning route of two different vehicles. With this operation the planning route of each vehicle can be altered and generate new possible solutions. The general MCSGA is described in Algorithm 1.

Algorithm 1. MCSGA Algorithm

```

1: Generate m clusters randomly
2: Best_Globale = Calculate_fitness( $CT_0$ )           \ \  $CT_0$  is the center of the first cluster
3: For each (i=1) in m-1:                           \ \ M is the number of clusters
4:   b = Calculate_fitness( $CT_i$ )                   \ \  $CT_i$  is the center of cluster i
5:   if Best_Globale > b :
6:     Best_Globale = b
7: endFor
8: Generate_initiale_population (PLi)
8: Best_solution_pop = Calculate_fitness (PL[0])     \ \ P[0] is the first solution in the population
9: For each (j=1) in (Pop_size - 1) :               \ \ Pop_size is the number of element of the population
10:  F = Calculate_fitness (PL[j])
11:  if Best_solution_pop > F:
12:    Best_solution_pop = F
13: endFor
14: While ( i=0) <  $I_{max}$  ) Do                        \ \  $I_{max}$  is the number maximum of iteration
15:  Selection_operator (PL)
16:  Crossover_operator(PL)
17:  Update_population
18:  Update(Best_solution_pop)
19:  For each (i=0) in m-2:
20:    if  $D(CT_i, Best\_solution\_pop) < D(CT_{i+1}, Best\_solution\_pop)$  : \ \  $D(a,b)$  is the distance between solution a and b
21:      if Cluster_size(i) <  $CA_{max}$ :                 \ \ Cluster_size is the number of elements in cluster i
22:        Assign Best_Solution_pop to cluster i
23:        Update the center  $C_i$  and increase the volume of cluster i
24:        Update (Best_Globale)
25:      else
26:        Reduce the volume of cluster i to zero
27:        if ( $r_{max} < r_i$ ):
28:          Local_Search ( $CT_i$ )
29:          Update the center  $CT_i$  and Increase  $r_i$  by one
30:          Update(Best_Globale)
31:        else
32:          Mutation_operator( $CT_i$ )
33:          Update the center  $CT_i$  and reduce  $r_i$  to zero
34:          Update(Best_Globale)
35:        endif
36:      endif
37:    else
38:      if Cluster_size(i+1) <  $CA_{max}$ 
39:        Assign Best_Solution_pop to cluster i+1
40:        Update the center  $CT_{i+1}$  and increase the volume of the cluster i + 1
41:        Update (Best_Globale)
42:      else:
43:        Reduce the volume of the Cluster i+1 to zero
44:        if ( $r_{max} < r_{i+1}$ ):
45:          Local_Search ( $CT_{i+1}$ )
46:          Update the center  $C_{i+1}$  and Increase  $r_{i+1}$  by one
47:          Update(Best_Globale)
48:        else
49:          Mutation_operator( $CT_{i+1}$ )
50:          Update the center  $CT_{i+1}$  and reduce  $r_i$  to zero
51:          Update(Best_Globale)
52:        Endif
53:      endif
54:    endif
55:  endfor
56: endwhile
57: Return(Best_Globale)

```

5. Computational experiments

In this section, two computational experiments studies are presented. The first one evaluates the effectiveness of the proposed MCSGA approach and the second one analyzes the impact of the travel time uncertainty of the vehicles on the quality of solutions.

5.1 Parameter Setting Selection

Parameter settings are a primordial factor to improve the performance of metaheuristics. The number of clusters (CL), the size of population (SP), the crossover rate (CR), the number of iteration (I_{max}) and the number of candidate parents to mixed in multi-parent crossover operation (M) are the five primordial parameters in the proposed MCSGA. To find a suitable combination of the five mentioned parameters, different preliminary experiments based on the Irace package are realized (López-Ibáñez et al., 2016). Irace uses a set of possible values for each parameter and compares the performance of the algorithm on training instances. Table 1 summarizes the results obtained for each parameter tuned in the MCSGA.

Table 1

Instances generated by Irace

Parameters-setting	Tested values	Best value
The number of clusters (CL)	{20,25,30,40}	30
The size of population (SP)	{1000,1500,2000,2500}	2000
the crossover rate (CR)	{0.5,0.6,0.8,0.9}	0.9
the number of iteration I_{max}	{150,200,250,300}	300
the number of candidate parents to mixed in multi-parent crossover operation (M)	{3,4,5,6}	4

5.2 Instances

In our experiments, the benchmark instances from (Schneider et al., 2014) are used. These instances are divided into three types: instances with random customer distribution (R), instances with clustered customer distribution (C), and instances with a mixture of both (RC).

Table 2

Comparison between the results obtained with CPLEX and MCSGA for the deterministic model

Instances	Type of instances	CPLEX		MCSGA	
		OV	TR	AOV	TR
c101C5	C	129.00	10	129.00	4
c101C10	C	185.23	16	185.23	12
c103C5	C	116.34	5	116.34	8
c103C15	C	209.75	2830	209.75	40
c104C10	C	166.08	24	166.08	21
c106C15	C	195.26	1984	195.26	32
c202C10	C	161.48	63	161.48	15
c202C15	C	210.517	3801	210.517	49
c205C10	C	159.24	34	159.24	13
c206C5	C	127.72	18	127.72	4
c208C5	C	116.15	12	116.15	5
c208C15	C	199.081	1934	199.081	53
r102C10	R	162.37	120	162.37	22
r102C15	R	218.72	3789	218.72	48
r103C10	R	155.43	259	155.43	15
r104C5	R	110.50	34	110.50	4
r105C5	R	113.41	22	113.41	6
r105C15	R	204.42	3439	204.42	38
r201C10	R	161.23	282	161.23	21
r202C5	R	128.88	27	128.88	6
r202C15	R	208.30	2892	208.30	59
r203C5	R	116.85	40	116.85	5
r203C10	R	158.40	419	158.40	39
r209C15	R	197.98	3831	197.98	69
rc102C10	RC	188.53	820	188.53	28
rc103C15	RC	214.61	6391	214.61	66
rc105C5	RC	121.88	40	121.88	3
rc108C5	RC	129.58	20	129.58	3
rc108C10	RC	177.18	901	177.18	16
rc108C15	RC	210.75	6199	210.75	58
rc201C10	RC	171.51	801	171.51	18
rc202C15	RC	214.38	6830	214.38	66
rc204C5	RC	119.27	39	119.27	4
rc204C15	RC	200.58	6910	200.58	59
rc205C10	RC	173.90	728	173.90	19
rc208C5	RC	116.69	40	116.69	4

OV denotes the objective value, AOV denotes the average of the objective value after 30 runs and TR denotes the total run-time in seconds. The maximum duration for CPLEX was set to two hours.

To test the proposed metaheuristic and analyse the impact of travel time uncertainty of vehicles on the quality of solutions, it is necessary to know the optimal value or the upper bound of each instance used in the study. Therefore, just the instances on which we can obtain an optimal or an upper bound with CPLEX version 20.1.0 in a maximum running time of 2 hours are selected. 1080 experiences for 36 instances are realized. Where, each of those instances are solved using the proposed MCSGA approach with 30 runs.

5.3 Performance of the proposed MCSGA approach on tested instances

In this section, the computational results obtained by the MCSGA algorithm were compared, firstly, with those obtained by CPLEX optimizer for the deterministic model. The deterministic model is the model presented in section 3 but without the constraints that consider the travel time uncertainty of the vehicles and, secondly, with both CPLEX optimizer and the classical CSA algorithm for the under-travel time uncertainty model formulation presented in section 3. To consider travel time uncertainty of vehicles, different scenarios and different probabilities of occurrence of these scenarios are used. In instance with five customers one scenario is used, in instance with ten customers, 2 scenarios are used and in instances with 15 or more customers, 3 scenarios are tested. The probabilities of occurrence considered for each scenario k are $P_k=0$, $P_k=0.5$ and $P_k=1, (\forall k \in w)$. This value of P_k has been selected randomly. The proposed MCSGA is implemented with python language on DELL computer with an Intel core i7, 8GB of ram memory. Table 2 provides the results found with CPLEX and MCSGA for deterministic models. The first two columns present the name and the type of instances respectively, column three contains the objective value and the running time obtained by CPLEX and the last column contains the average objective value from 30 runs and the running time obtained by MCSGA. According to the results presented in Table 2, we conclude that the MCSGA method can obtain the optimal solution for all instances tested after 30 runs. We observe that in all instances tested the optimal solution is achieved in each run among 30 runs tested except in “rc108C15”, “rc204C15” and “rc103C15”, This is because when the size of customers increases the optimal solution became hard to achieve. At the level of running time, the proposed MCSGA can obtain optimal solutions in a very short running time for all instances while CPLEX take approximately one hour in most of the case and two hours in some tested instances as “c103C15” and “rc108C15”.

Table 3

Comparison between the results obtained with CPLEX and MCSGA for under uncertain travel time model

Instances	scenarios	CPLEX (Under uncertain travel time model)						MCSGA (Under uncertain travel time model)					
		$P_k=0$		$P_k=0.5$		$P_k=1$		$P_k=0$		$P_k=0.5$		$P_k=1$	
		OV	TR	OV	TR	OV	TR	AOV	TR	AOV	TR	AOV	TR
c101C5	1	137.50	10	147.00	11	148.50	17	137.50	3	147.00	3	148.50	4
c101C10	2	195.96	18	196.62	21	201.62	24	195.96	8	196.62	9	201.62	9
c103C5	1	125.07	7	133.07	8	133.07	8	125.07	5	133.07	6	133.07	7
c103C15	3	224.12	2854	224.27	2954	242.27	3001	224.12	43	224.27	55	242.27	73
c104C10	2	176.99	29	176.99	31	181.99	29	176.99	14	176.99	18	181.99	18
c106C15	3	216.05	3002	217.30	3012	232.55	3013	216.05	53	217.30	59	232.55	59
c202C10	2	172.52	30	172.52	33	177.52	33	172.52	12	172.52	12	177.52	13
c202C15	3	224.51	2963	223.01	2892	239.51	2983	224.51	69	223.01	72	239.51	75
c205C10	2	172.95	44	175.79	59	180.79	62	172.95	10	175.79	13	180.79	21
c206C5	1	136.89	13	144.89	20	144.89	18	136.89	4	144.89	4	144.89	4
c208C5	1	124.83	15	132.83	16	132.83	15	124.83	3	132.83	5	132.83	5
c208C15	2	213.93	4023	213.93	5039	229.34	4034	213.93	70	213.93	88	229.34	101
r102C10	2	174.75	38	174.75	30	179.75	33	174.75	12	174.75	16	179.75	18
r102C15	3	233.24	3980	232.68	5090	249.93	6040	233.24	54	232.68	59	249.93	53
r103C10	2	176.60	56	176.60	66	181.60	60	176.60	15	176.60	19	181.60	18
r104C5	1	121.03	13	129.48	11	129.48	14	121.03	4	129.48	4	129.48	6
r105C5	1	120.99	11	128.99	11	128.99	12	120.99	6	128.99	9	128.99	16
r105C15	3	218.79	3698	218.79	6049	233.79	7890	218.79	76	218.79	93	233.79	106
r201C10	2	171.87	36	171.87	53	176.87	56	171.87	14	171.87	19	176.87	34
r202C5	1	121.43	12	128.99	10	130.48	14	121.43	3	128.99	7	130.48	8
r202C15	3	225.50	4566	225.00	6903	240.50	8045	225.50	52	225.00	60	240.50	78
r203C5	1	127.31	13	135.31	10	135.31	18	127.31	4	135.31	4	135.31	4
r203C10	2	169.10	49	169.59	22	174.59	40	169.10	15	169.59	20	174.59	28
r209C15	3	213.18	4101	213.19	6755	231.32	8094	213.18	54	213.19	67	231.32	69
rc102C10	2	199.73	46	199.73	73	204.73	89	199.73	18	199.73	22	204.73	40
rc103C15	3	229.53	3679	229.14	5002	247.10	7094	229.53	59	229.14	77	247.10	75
rc105C5	1	131.64	10	140.57	10	141.56	16	131.64	4	140.57	4	141.56	4
rc108C5	1	138.73	14	147.53	12	148.47	12	138.73	6	147.53	10	148.47	13
rc108C10	2	188.48	66	188.48	68	193.48	79	188.48	13	188.48	20	193.48	22
rc108C15	3	226.58	3299	226.17	3902	242.67	5134	226.58	73	226.17	96	242.67	102
rc201C10	2	182.28	34	182.28	63	187.28	66	182.28	14	182.28	28	187.28	38
rc202C15	3	230.82	4258	229.94	6109	247.61	8956	230.82	50	229.94	75	247.61	78
rc204C5	1	128.85	10	147.53	11	148.47	15	128.85	3	147.53	8	148.47	10
rc204C15	3	215.25	4390	215.36	8944	231.86	9805	215.25	49	215.36	66	231.86	70
rc205C10	2	185.81	49	185.81	79	190.81	80	185.81	11	185.81	18	190.81	20
rc208C5	1	126.15	11	134.64	11	135.27	12	126.15	5	134.64	5	135.27	6

OV denotes the objective value, AOV denotes the average of the objective value after 30 runs, TR denotes the total run-time in seconds and P_k denotes the probability of occurrence of scenario k ($\forall k \in w$). The maximum duration for CPLEX was set to two hours.

Table 4

Comparison between the results obtained with Classical CSA and MCSGA for under uncertain travel time model

Instances	scenarios	CSA(Under uncertain travel time model)						MCSGA (Under uncertain travel time model)					
		$P_k=0$		$P_k=0.5$		$P_k=1$		$P_k=0$		$P_k=0.5$		$P_k=1$	
		OV	TR	OV	TR	OV	TR	AOV	TR	AOV	TR	AOV	TR
c101C5	1	137,50	3	147,00	4	148,50	3	137,50	3	147,00	3	148,50	4
c101C10	2	195,96	11	197,53	23	209,34	44	195,96	8	196,62	9	201,62	9
c103C5	1	125,07	4	133,07	7	133,07	6	125,07	5	133,07	6	133,07	7
c103C15	3	265,12	68	251,47	80	260,24	99	224,12	43	224,27	55	242,27	73
c104C10	2	183,99	13	176,99	24	181,99	56	176,99	14	176,99	18	181,99	18
c106C15	3	266,05	44	223,45	77	243,34	83	216,05	53	217,30	59	232,55	59
c202C10	2	177,54	31	172,52	53	177,52	86	172,52	12	172,52	12	177,52	13
c202C15	3	259,67	68	264,45	79	266,56	83	224,51	69	223,01	72	239,51	75
c205C10	2	174,77	12	175,79	25	180,79	62	172,95	10	175,79	13	180,79	21
c206C5	1	136,89	5	144,89	8	144,89	7	136,89	4	144,89	4	144,89	4
c208C5	1	124,83	4	132,83	5	132,83	6	124,83	3	132,83	5	132,83	5
c208C15	2	229,32	63	224,23	79	243,56	94	213,93	70	213,93	88	229,34	101
r102C10	2	174,75	18	174,75	30	179,75	33	174,75	12	174,75	16	179,75	18
r102C15	3	260,34	45	254,56	80	256,55	89	233,24	54	232,68	59	249,93	53
r103C10	2	176,60	21	176,60	26	181,60	50	176,60	15	176,60	19	181,60	18
r104C5	1	121,03	4	129,48	6	129,48	8	121,03	4	129,48	4	129,48	6
r105C5	1	120,99	5	128,99	7	128,99	9	120,99	6	128,99	9	128,99	16
r105C15	3	245,56	63	232,45	69	255,59	70	218,79	76	218,79	93	233,79	106
r201C10	2	174,54	23	171,87	23	176,87	36	171,87	14	171,87	19	176,87	34
r202C5	1	121,43	4	128,99	5	130,48	7	121,43	3	128,99	7	130,48	8
r202C15	3	234,34	65	235,10	69	230,50	80	225,50	52	225,00	60	240,50	78
r203C5	1	127,31	6	135,31	4	135,31	5	127,31	4	135,31	4	135,31	4
r203C10	2	178,10	32	169,59	26	174,59	34	169,10	15	169,59	20	174,59	28
r209C15	3	232,33	65	252,23	67	267,21	80	213,18	54	213,19	67	231,32	69
rc102C10	2	199,73	25	199,73	33	204,73	44	199,73	18	199,73	22	204,73	40
rc103C15	3	234,55	59	256,33	50	25,10	70	229,53	59	229,14	77	247,10	75
rc105C5	1	131,64	5	140,57	7	141,56	6	131,64	4	140,57	4	141,56	4
rc108C5	1	138,73	6	147,53	6	148,47	5	138,73	6	147,53	10	148,47	13
rc108C10	2	188,48	24	188,48	28	193,48	29	188,48	13	188,48	20	193,48	22
rc108C15	3	235,45	56	245,15	93	256,56	98	226,58	73	226,17	96	242,67	102
rc201C10	2	182,28	22	182,28	23	187,28	34	182,28	14	182,28	28	187,28	38
rc202C15	3	233,45	44	230,45	61	257,31	89	230,82	50	229,94	75	247,61	78
rc204C5	1	128,85	4	147,53	5	148,47	5	128,85	3	147,53	8	148,47	10
rc204C15	3	216,45	64	216,56	76	242,45	85	215,25	49	215,36	66	231,86	70
rc205C10	2	185,81	29	185,81	25	190,81	39	185,81	11	185,81	18	190,81	20
rc208C5	1	126,15	5	134,64	5	135,27	4	126,15	5	134,64	5	135,27	6

OV denotes the objective value, AOV denotes the average of the objective value after 30 runs, TR denotes the total run-time in seconds and P_k denotes the probability of occurrence of scenario k ($\forall k \in \omega$).

Table 3 and Table 4 present the results found with CPLEX, CSA (Chaves & Lorena, 2010) and MCSGA for the under-travel time uncertainty model. In table 3, the first column presents the name of instance, the second column presents the number of scenarios tested in each instance, the column three provide the objective value and the running time obtained by CPLEX for each probability tested ($P_k=0$, $P_k=0.5$ and $P_k=1$) and the last column provide the objective value and the running time obtained by the MCSGA for each probability tested ($P_k=0$, $P_k=0.5$ and $P_k=1$). In Table 4, the first column presents the name of instance, the second column presents the number of scenarios tested in each instance, the column three provide the objective value and the running time obtained by the CSA for each probability tested ($P_k=0$, $P_k=0.5$ and $P_k=1$) and the last column presents the objective value and the running time obtained by the MCSGA for each probability tested ($P_k=0$, $P_k=0.5$ and $P_k=1$).

Table 3 and 4 show that both, the proposed MCSGA metaheuristic and CPLEX are able to find the optimal solution for the majority of the instances tested whatever the probability of the scenarios. This is not the case for CSA which is unable to find optimal solutions for the instances with 15 customers. The execution times achieved by the MCSGA method are much shorter than the times achieved by CPLEX, especially when 'P=1' and the number of customers is high. According to the tables 2, 3 and 4 we notice that the values of the objective function in under travel time uncertainty model are relatively high in comparison with those of deterministic model. However, with U-EVRWT model we obtain a more flexible route planning that avoids any disruptions in initial planning which is not possible in deterministic model.

5.4 Analyzing the impact of travel time uncertainty of vehicles on results

In this section, we analyze the impact of the travel time uncertainty of vehicles on three important components of the final solution which are the number of vehicles used in each routing plan, the objective function values and the running time. Fig.

6, Fig. 7 and Fig. 8 show the impact of travel time uncertainty on: the number of vehicles used in each routing plan, the objective function values and the running time consecutively. According to the figures, we notice that the average number of vehicles, the running time of the algorithm and the objective function values in U-EVRWT model is relatively high in comparison with the average number of vehicles, the running time of the algorithm and the objective function values in deterministic model, consequently, we conclude that, the general cost of vehicles routing plan in U-EVRWT model can be high in comparison with deterministic one, this is correct especially when the number of customers is high. However, we should note that this conclusion is valid just in absence of unexpected events. In cases where the unexpected events and travel time uncertainty are present, the deterministic model returns infeasible solutions while the U-EVRWT returns feasible and efficient solutions.

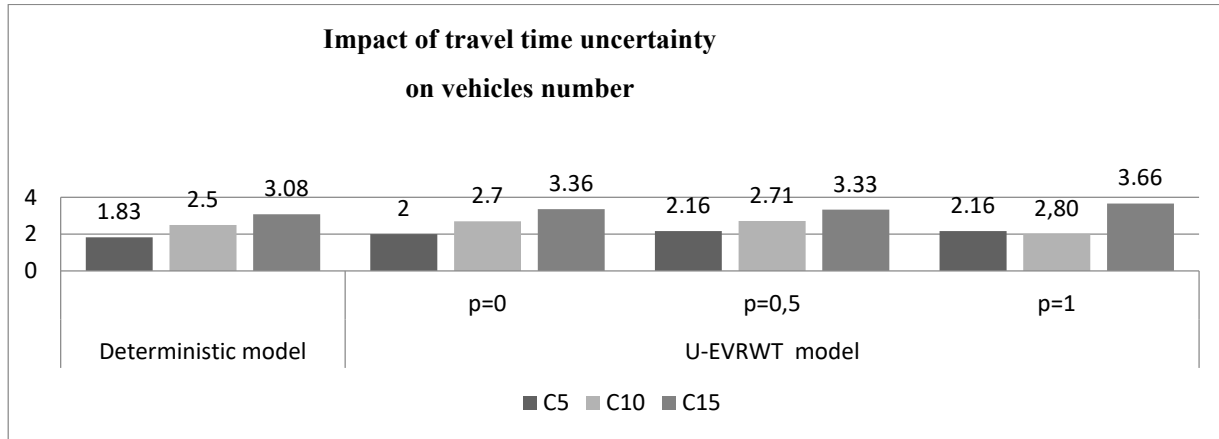


Fig. 6. Impact of travel time uncertainty on the vehicles number

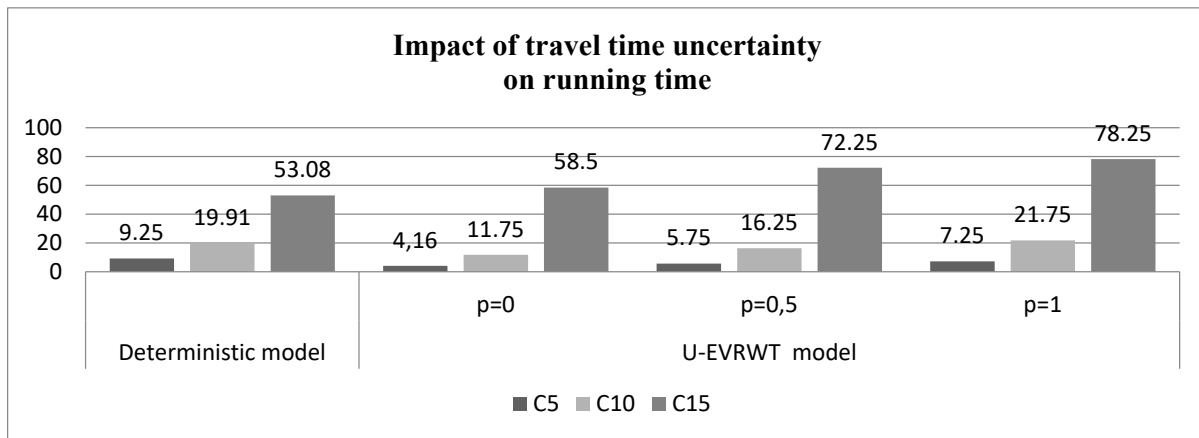


Fig. 7. Impact of travel time uncertainty on Running Time

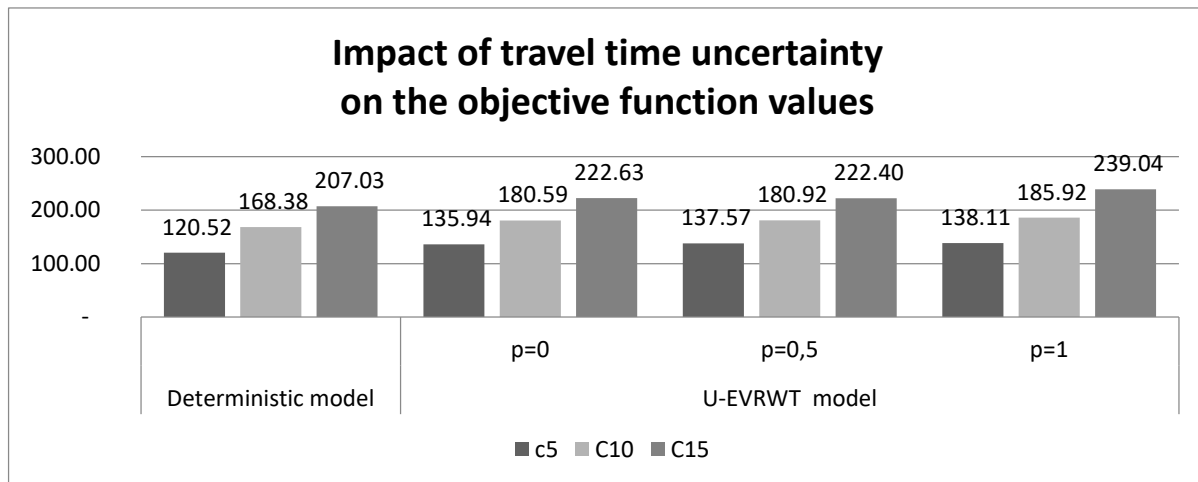


Fig. 8. Impact of travel time uncertainty on the objective function values

6. Conclusion

In this work an electric vehicle routing problem with time windows and under travel time uncertainty (U-EVRW) is addressed. A mixed integer mathematical model was proposed to solve the problem and find the optimal proactive routing plan of the electric vehicles under the travel time uncertainty. A MCSGA was developed as a new effective metaheuristic to solve the U-EVRW. Comparison results have indicated that the proposed MCSGA outperform the CPLEX optimizer and CSA in both, solution quality and running time. In addition, the computational experiences have shown the impact of the travel time uncertainty on the average number of vehicles, the running time of the algorithm and the objective function values in U-EVRW model. In the future working on U-EVRW a reactive strategy to solve the problem can be developed. Another possible case of study is the development of an exact solution approach to solve the problem.

References

- Afroditi, A., Boile, M., Theofanis, S., Sdoukopoulos, E., & Margaritis, D. (2014). Electric Vehicle Routing Problem with Industry Constraints: Trends and Insights for Future Research. *Transportation Research Procedia*, 3, 452–459. <https://doi.org/10.1016/j.trpro.2014.10.026>
- Allahviranloo, M., Chow, J. Y. J., & Recker, W. W. (2014). Selective vehicle routing problems under uncertainty without recourse. *Transportation Research Part E: Logistics and Transportation Review*, 62, 68–88. <https://doi.org/10.1016/j.tre.2013.12.004>
- Cao, E., Gao, R., & Lai, M. (2018). Research on the vehicle routing problem with interval demands. *Applied Mathematical Modelling*, 54, 332–346. <https://doi.org/10.1016/j.apm.2017.09.050>
- Chaves, A. A., & Lorena, L. A. N. (2010). Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, 37(3), Article 3. <https://doi.org/10.1016/j.cor.2008.09.011>
- Cortés-Murcia, D. L., Prodhon, C., & Murat Afsar, H. (2019). The electric vehicle routing problem with time windows, partial recharges and satellite customers. *Transportation Research Part E: Logistics and Transportation Review*, 130, 184–206. <https://doi.org/10.1016/j.tre.2019.08.015>
- Erdoğan, S., & Miller-Hooks, E. (2012). A Green Vehicle Routing Problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1), 100–114. <https://doi.org/10.1016/j.tre.2011.08.001>
- Hiermann, G., Hartl, R. F., Puchinger, J., & Vidal, T. (2019). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1), 235–248. <https://doi.org/10.1016/j.ejor.2018.06.025>
- Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations. *European Journal of Operational Research*, 252(3), 995–1018. <https://doi.org/10.1016/j.ejor.2016.01.038>
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* by Holland, John H - AbeBooks (M1 ed.). University of Michigan Press, Ann Arbor. <https://www.abebooks.com/book-search/isbn/9780472084609/>
- Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65, 111–127. <https://doi.org/10.1016/j.trc.2016.01.013>
- Koç, Ç., Jabali, O., Mendoza, J. E., & Laporte, G. (2019). The electric vehicle routing problem with shared charging stations. *International Transactions in Operational Research*, 26(4), 1211–1243. <https://doi.org/10.1111/itor.12620>
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43–58. <https://doi.org/10.1016/j.orp.2016.09.002>
- Macrina, G., Di Puglia Pugliese, L., Guerriero, F., & Laporte, G. (2019). The green mixed fleet vehicle routing problem with partial battery recharging and time windows. *Computers & Operations Research*, 101, 183–199. <https://doi.org/10.1016/j.cor.2018.07.012>
- Mao, H., Shi, J., Zhou, Y., & Zhang, G. (2020). The Electric Vehicle Routing Problem With Time Windows and Multiple Recharging Options. *IEEE Access*, 8, 114864–114875. <https://doi.org/10.1109/ACCESS.2020.3003000>
- Messaoud, E. (2021). A chance constrained programming model and an improved large neighborhood search algorithm for the electric vehicle routing problem with stochastic travel times. *Evolutionary Intelligence*. <https://doi.org/10.1007/s12065-021-00648-0>
- Montoya, A., Guéret, C., Mendoza, J. E., & Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103, 87–110. <https://doi.org/10.1016/j.trb.2017.02.004>
- Pan, W., & Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, 53(1), 405–422. <https://doi.org/10.1007/s10489-022-03456-w>
- Pelletier, S., Jabali, O., & Laporte, G. (2019). The electric vehicle routing problem with energy consumption uncertainty. *Transportation Research Part B: Methodological*, 126, 225–255. <https://doi.org/10.1016/j.trb.2019.06.006>
- Qin, H., Su, X., Ren, T., & Luo, Z. (2021). A review on the electric vehicle routing problems: Variants and algorithms. *Frontiers of Engineering Management*, 8(3), 370–389. <https://doi.org/10.1007/s42524-021-0157-1>

- Schneider, M., Stenger, A., & Goetze, D. (2014). The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transportation Science*, 48(4), 500–520. <https://doi.org/10.1287/trsc.2013.0490>
- Poalo Toth et Daniele Vigo(2014). *Vehicle routing problems, methods, and application Philadelphia: Mathematical Optimization Society*
- Verma, A. (2018). Electric vehicle routing problem with time windows, recharging stations and battery swapping stations. *EURO Journal on Transportation and Logistics*, 7(4), 415–451. <https://doi.org/10.1007/s13676-018-0136-9>
- Wang, Y., Assogba, K., Fan, J., Xu, M., Liu, Y., & Wang, H. (2019). Multi-depot green vehicle routing problem with shared transportation resource: Integration of time-dependent speed and piecewise penalty cost. *Journal of Cleaner Production*, 232, 12–29. <https://doi.org/10.1016/j.jclepro.2019.05.344>
- Zhang, X., & Tang, L. (2009). A new hybrid ant colony optimization algorithm for the vehicle routing problem. *Pattern Recognition Letters*, 30(9), 848–855. <https://doi.org/10.1016/j.patrec.2008.06.001>
- Zhao, M., & Lu, Y. (2019). A Heuristic Approach for a Real-World Electric Vehicle Routing Problem. *Algorithms*, 12(2), Article 2. <https://doi.org/10.3390/a12020045>



© 2023 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).