International Journal of Applied Operational Research
Vol. 4, No. 3, pp. 49-64, Summer 2014

Journal homepage: www.ijorlu.ir

A Hybrid Metaheuristic Algorithm for the Vehicle Routing Problem with Delivery Time Cost

A. Kourank Beheshti*, S.R. Hejazi, S. H. Mirmohammadi

Received: 29 January 2014 **;** **Accepted:** 5 May 2014

Abstract This paper addresses the Vehicle Routing Problem with Delivery Time Cost. This problem aims to find a set of routes of minimal total costs including the travelling cost and delivery time cost, starting and ending at the depot, in such a way that each customer is visited by one vehicle given the capacity of the vehicle to satisfy a specific demand. In this research, a hybrid metaheuristic approach based on Electromagnetism and simulated annealing algorithms is proposed. The purpose of such a combination is to have the benefits of both algorithms. Simulated Annealing (SA) algorithm is powerful in escaping from local optimums. On the other hand, Electromagnetism (EM) algorithm generates wide varieties of solution populations. The computational results of some instances are reported and then these results are compared with the lower bound of problem. The results demonstrate the effectiveness of the metaheuristic algorithm in solving this model.

Keywords: The Vehicle Routing Problem With Delivery Time Cost, Simulated Annealing, Electromagnetism, Hybrid Metaheuristic.

1 Introduction

Customer service level is of prime importance in today's competitive world and has various dimensions with delivery quality being one of the most important ones. It affects both customer satisfaction and distribution costs. Delivery quality has several parameters such as deliver time window options, time window size, etc. The following sections of this paper focus on one of these parameters, namely delivery time setting.

A Time Window (TW) is defined as the time interval within which a vehicle has to arrive at a node, and it is usually characterized by an early arrival time (EAT) and a late arrival time (LAT)[1]. The time window can be specified in terms of being either a single-sided or a double-sided window. In a single-sided time window, the pickup points usually specify the deadlines by which they must be serviced. In double-sided time window, however, both the

* **Corresponding Author.** (✉)
E-mail: a.korankbeheshti@in.iut.ac.ir (A.K.Beheshti)

A. Kourank Beheshti
Ph.D. Candidate, Department of Industrial & Systems Engineering, Isfahan University of Technology, Isfahan, Iran

S.R. Hejazi
Associate professor, Department of Industrial & Systems Engineering, Isfahan University of Technology, Isfahan, Iran

S. H. Mirmohammadi
Assistant professor, Department of Industrial & Systems Engineering, Isfahan University of Technology, Isfahan, Iran

[Downloaded from mail.ijaor.ir on 2026-11-27]

earliest and the latest service times are imposed by the nodes[2]. In fact, in double-sided time window, EAT is equal to LAT.

As mentioned earlier, customer service level affects both customer satisfaction and delivery costs. Consequently, delivery time must be set by taking into account both customer satisfaction and distribution costs. Distribution costs include such costs such as fuel cost, vehicle cost, driver salary, etc. These costs are usually considered in vehicle routing problem (VRP) models. VRP is one of the most applicable problems in industries and plays a pivotal role in logistics. Vehicle routing problem with time window (VRPTW) is an extension of the VRP that considers service quality; in this model, delivery locations have predefined time windows, within which the deliveries (or visits) must be made.

In the present paper, an integrated mathematical model and a hybrid metaheuristic algorithm are proposed for the vehicle routing problem with delivery time cost. In the second section, the related literature is briefly reviewed. This is followed by a presentation of the model in section three and the hybrid metaheuristic in section four. The computational results of the study are discussed in section five and the paper ends with the concluding remarks in the section six.

2 Literature Review

The Vehicle Routing Problem (VRP) is a well-known combinatorial optimization problem (COP) and has a wide range of applications. It deals with determination of the minimum cost routes from a central depot to a set of geographically dispersed customers. The Vehicle Routing Problem with Time Window (VRPTW) is an extension of the VRP where a constraint is added requiring the start of service at each customer within a time window. If the time window constraints must be satisfied strictly, such a problem is called the Vehicle routing problem with hard time window (abbreviated as VRPHTW). The Vehicle routing problem with soft time window (abbreviated as VRPSTW) is a relaxation of the VRPHTW. Time windows in the former can be violated if a penalty is paid and this penalty is often assumed linearly proportional to the degree of violation, but in the latter violations are infeasible. In the VRPSTW, it is usually assumed that customers determine a time window and the penalty for the time window constraint is linearly to its earliness and tardiness [3-5]. Some researchers have studied the VRPSTW that considers penalties only on late arrivals while waiting on early arrivals is allowed without any cost, the condition which is referred to as the Vehicle Routing Problem with Semi Soft Time Windows (VRPSSTW)[6, 7]. In some cases, violation of time windows does not directly incur any penalty cost, although the satisfaction levels of customers (the service level of suppliers) may drop and lead to profit loss in the long term. In these cases, researchers usually apply a fuzzy theory to the routing problem ([8-10]). The vehicle routing problem with multiple time windows (abbreviated as VRPMTW) have also been considered[11, 12].

Ibaraki et al. [13] proposed local search algorithms for the vehicle routing problem with soft time window constraints. The time window constraint for each customer was treated as a penalty function, which was very general in the sense that it could be non-convex and discontinuous as long as it was piecewise linear. They used local search to assign customers to vehicles and to find orders of customers for vehicles to visit. After fixing the order of customers for a vehicle to visit, they proposed a dynamic programming algorithm to efficiently compute the optimal start times of services for customers in a given route. Ibaraki et al. [14] treated the time window constraint for each customer as a penalty function, and

[Downloaded from mail.ijaor.ir on 2026-11-27]

assumed that it was convex and piecewise linear. Given an order of customers each vehicle to visit, dynamic programming was used to determine the optimal start time to serve the customers so that the total time penalty was minimized.

However, in this paper delivery time set with regard to routing and delivery time costs. This approach can be used when information of all customers such as demand are known. Consequently, it cannot be utilized in some e-fulfillment services where delivery time must be immediately determined in customer call time. However, in other services, it can be used and can determine optimal delivery time window. In the following section, a mathematical model of the current problem is presented and owing to the complexity of the model, a hybrid metaheuristic algorithm is presented.

3 Mathematical Model

In this section, a mathematical model is presented for the vehicle routing problem with delivery time cost (VRPDTC).
The assumptions of this model are summarized as follows:

1. A single depot and a homogeneous fleet are considered.
2. A single product for distribution is considered.
3. Planning is worked out for one day.
4. For each customer a delivery time cost function is defined. These can be the same for all customers.
5. Each customer is visited by exactly one vehicle.
6. Customers accept the delivery time that the distributer has set.
7. No disruptions during travel might occur due to weather, human, or other unexpected factors.

This problem is defined as: “given a set of customers, a set of vehicles and a depot, the VRPDTC is to find a set of routes of minimal total cost, starting and ending at the depot, in such a way that each customer is visited by one vehicle with regard to the capacity of the vehicle to satisfy a specific demand”. Total cost includes fuel costs and delivery time costs.
More formally, this problem is defined on a directed graph G = (V,A), where $V = \{v_0,...,v_n\}$ is the set of nodes and A is the set of arcs. Vertex v_0 is a depot node; other vertexes are customer nodes. Non-negative travel time t_{ij} is associated with each arc(v_i,v_j), which satisfies the triangle inequality. Each customer $v_i \in V \setminus \{v_0\}$ has a demand d_i. Furthermore, let γ and δ_i be the per time unit cost for fuel and delivery time, respectively. A homogeneous fleet of vehicle with capacity q is available to server the customers.
Parameters and decision variables of the mathematical model are displayed in Table 1.

[Downloaded from mail.ijaor.ir on 2026-11-27]

Table 1 The parameters and decision variables of the proposed model

Parameter/ Decision Variables	Definition
K	Maximum number of vehicles
q	Capacity of each vehicle
t_{ij}	Traveling time between two vertices $(i,j) \in A$
d_i	The demand of customer *i*
γ	Fuel cost of a vehicle per time unit
δ_i	Delivery time cost per time unit of customer i
x_{ijk}	This variable is equal to 1 if arc *(i,j)* is used by vehicle *k* and 0 otherwise
s_{ik}	The start time of service for customer *i* when serviced by vehicle *k*
a_i	The delivery time of customer *i*

The problem can then be described using the following model:

$$Min \sum_{k\in K}\sum_{i\in V}\sum_{j\in V}\gamma t_{ij}x_{ijk} + \sum_{i\in V} f(a_i) \quad (1)$$

$$\sum_{k\in K}\sum_{j\in V} x_{ijk} = 1 \qquad \forall i \in V \quad (2)$$

$$\sum_{i\in V} d_i \sum_{j\in V} x_{ijk} \le q \qquad \forall k \in K \quad (3)$$

$$\sum_{j\in V} x_{0jk} = 1 \qquad \forall k \in K \quad (4)$$

$$\sum_{i\in V} x_{ihk} - \sum_{j\in V} x_{hjk} = 0 \qquad \forall h \in V, k \in K \quad (5)$$

$$s_{ik} + t_{ij} - M(1 - x_{ijk}) \le s_{jk} \qquad \forall i,j \in V, \forall k \in K \quad (6)$$

$$a_i = \sum_{k\in K}\sum_{j\in V} s_{ik}x_{jik} \qquad \forall i \in V \quad (7)$$

$$x_{ijk} \in \{0,1\} \qquad \forall i,j \in V, \forall k \in K \quad (8)$$

$$s_{ik} \ge 0 \qquad \forall i \in V, \forall k \in K \quad (9)$$

$$a_i \ge 0 \qquad \forall i \in V \quad (10)$$

The objective function (1) of this model is to minimize the total cost including fuel and delivery time costs. $f(a_i)$ is a delivery time cost function. We considered a simple delivery time cost function and assumed that $f(a_i) = \delta_i a_i$ in this study, however delivery time assignment can has various functions such as earliness/tardiness penalty function in VRPSTW and general piecewise cost functions in real situation. Constraint (2) ensures that each customer exactly is met by one vehicle. Constraint (3) is the guarantee capacity constraint. Next, constraints (4-5), characterize the flow on the path to be followed by vehicle k. Constraint (6) states that vehicle k cannot arrive at j before $s_{ik} + t_{ij}$, if it is traveling from i to j. Here M is a large scalar. Note that this constraint also forbids subtours in the solution. Constraint (7) calculates the start time of the service at node i. We can remove it and use $\sum_{k\in K}\sum_{j\in V} s_{ik}x_{jik}$ instead of a_i in the objective function. Constraints (8-10) are the definition

[Downloaded from mail.ijaor.ir on 2026-11-27]

constraints of the decision variables and they refer to whether the vehicle k moves from vertex i to j or no, the start time of the service at customer i when serviced by vehicle k, and the delivery start time of customer i, respectively.

It can use the exact algorithm to solve the small and relatively medium size problems; however, to solve the large size problems, given the complexity of the model, it should employ heuristic and metaheuristic algorithms. Hence, in the following section, to solve the problems, a metaheuristic approach based on Electromagnetism algorithm is proposed.

4 Metaheuristic Algorithm

In this paper, hybrid Parallel Simulated Annealing-Electromagnetism algorithm is proposed to solve the large size problems. The purpose of such a combination is to have the benefits of both algorithms. Simulated Annealing (SA) algorithm is powerful in escaping from local optimums. On the other hand, Electromagnetism (EM) algorithm generates wide varieties of solution populations[15]. In this section, we first introduce SA and EM algorithms and then describe our proposed hybrid method in detail.

4.1 Parallel Simulated Annealing Algorithm

Parallel Simulated Annealing algorithm is a developed form of Simulated Annealing (SA) algorithm. SA is a metaheuristic method, which is capable of solving combinatorial optimization problems efficiently. In fact, SA is a probability method, which imitates the process of physical melting of solids in order to find the solution of the combinatorial problem.

The superiority of this algorithm lies in the fact that it can escape from local optimal points. However, one of its primary disadvantages is the enormous computational efforts required to reach good and effective solutions. This drawback stems from the mechanism adopted by this algorithm, since the algorithm is not population-based and starts searching from one point in the solution space. A way to overcome such a drawback is to utilize several SA processes in a parallel manner to search the solution space. Several methods exist to make SA algorithms parallel. For instance, Hiroyasu et al. [16] have presented one such paralleling method. In this method, the searching process begins from several points in the solution space and after some steps; the crossover operator of Genetic algorithm is used to generate good solutions.

4.2 Electromagnetism-like Algorithm

Electromagnetism-like (EM) algorithm was first introduced by Bribile and Fang[17]. Debels *et al*. [18]have implemented the hybrid Electromagnetism algorithm in solving combinatorial optimization problems and they were pioneer in using EM for solving a combinatorial optimization problem.

The EM-type algorithm, which takes advantage of the attraction–repulsion mechanism of electromagnetic theory, has been utilized for optimization problems. In fact, the EM algorithm is a new population-based meta-heuristic method simulating Coulomb's law. The approach starts with a random population of points (particles) from the feasible region. Each

[Downloaded from mail.ijaor.ir on 2026-11-27]

particle corresponds to a solution, and its charge represents its quality. In other words, in maximization (minimization) problems, a better solution has a higher (lower) charge. A better solution attracts other particles in order to converge to that point, while a bad solution pushes them away. The following equation represents the relation between the charge of particles and the objective function to be optimized:

$$q^i = \exp(-m\frac{f(x^i)-f(x^{best})}{\sum_{k=1}^{m}(f(x^k)-f(x^{best}))} \quad (11)$$

In this equation, q_i is the charge of particle i; $f(x^i)$, $f(x^{best})$, and $f(x^k)$ represents the objective functions value of particle i, the best result, and particle k, respectively. m is the population size.

Apparently, the charge of particle determines the magnitude of the force exerted on other points. Particles move in the resultant force F^i direction exerted on them. The following equation calculates the resultant force F^i :

$$F^i = \sum_{j\neq i}^{m}\left|x^i - x^j\right|\frac{q^i q^j}{\left\|x^j - x^i\right\|^2} \quad (12)$$

It can be seen that the force between two particles is inversely proportionate to the square of distance between the points and directly proportionate to particle charges.
The main methods of EM including determination of initial population, local search, and calculation of the motion force between particles and move them, are presented in Algorithm 1[15]:

Algorithm 1 - Electromagnetism

1. Initialize ()
2. **While** (has not met stop criterion) **do**
3. Local Search ()
4. Calculate total force F ()
5. Move particle by F ()
6. Evaluate particles ()
7. **End.**

4.3 Proposed Hybrid Method

The framework of the hybrid method consists of parallel SA and Electromagnetism-like algorithms. As noted before, Electromagnetism creates an excellent variation in the population and Parallel Simulated Annealing adds to the algorithm the power of escaping from local optimums. The flowchart of the proposed algorithm in this study is depicted in Fig.1. In this method, starting from the generated random points, n simulated annealing processes begin to search the solution space in a parallel manner. n is the population size. This operation begins with an initial temperature for all the processes. Then, in each process, the operation of searching is performed until an equilibrium state is reached (according to SA algorithm). This operation is carried out in each parallel process as follows: at first, a

[Downloaded from mail.ijaor.ir on 2026-11-27]

neighborhood solution is generated from the current solution and the value of its objective function is calculated. If the neighborhood solution is better than the current one, this solution replaces the current solution. Otherwise, according to equation (13), neighborhood solution is accepted with probability of p. In this equation Δf is the variation of the objective function values of current and neighborhood solutions and T is the current temperature of the process:

$$p = e^{-\frac{\Delta f}{T}} \tag{13}$$

The process of generating neighborhood continues until equilibrium is reached.
After all SA processes reach an equilibrium state, some current solutions of the processes move by Electromagnetism algorithm. In fact, good solutions attract other solutions and bad results repulse other solutions. After that, a local search is performed on the solutions. The process of lowering the temperature begins in the next stage and if the final condition (getting the final temperature) is not satisfied, simulated annealing processes start to operate from current solutions.

In this hybrid algorithm, some parameters and processes must be determined; these include the coding procedure of the problem solutions, the determination of the initial solutions, the initial temperature, the final temperature, the annealing process (lowering temperature), the equilibrium condition, the generation of neighborhoods, etc. In the following, these are depicted.

4.3.1 Coding Procedure of the Problem Solutions

As mentioned before, the electromagnetism algorithm is a meta-heuristic algorithm based on population developed to solve continuous problems efficiently. In this case, coding the solutions should be in the form of real numbers so that they can be used by the electromagnetism-like algorithm. To do so, the R-K approach is adopted. The concept of R-K is simple and can be easily applied. Suppose we have an array constructed from real numbers. Then by arranging them in descending or ascending order, we can have an order of numbers. Therefore, an array of real numbers corresponds to an order of numbers. Of course, in this method, in order to prevent the numbers from being dispersed, such real numbers must be defined in a specific interval. In fact, solutions are in the form of real numbers and anytime we want to calculate the objective function, such a transformation must be performed. This is displayed in Fig. 2. In this example, real numbers are defined on [0, 10].

4.3.2 Determination of the Initial Population

In the proposed method, the initial solutions of each process are randomly generated. Random numbers are generated in the range [l,u]. This generation range (GR)is set in the section 5-1.

[Downloaded from mail.ijaor.ir on 2026-11-27]

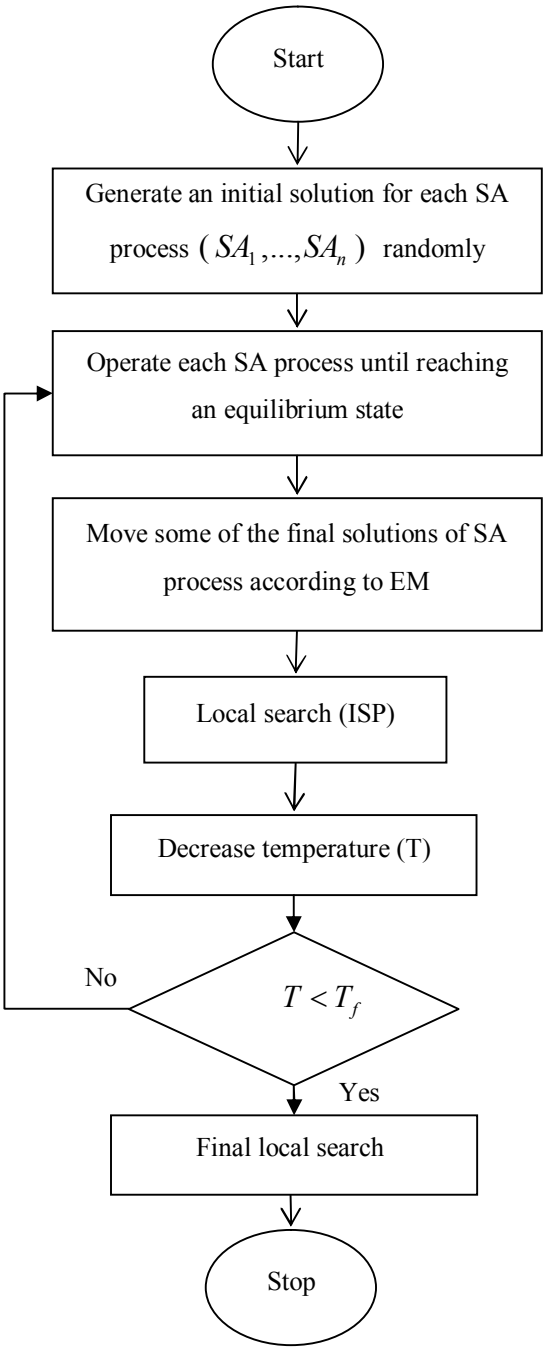

Fig. 1 Flowchart of Parallel-Electromagnetism Simulated Annealing

	1	2	3	4	5	6	7	8	9	10
Before:	0.5	9.6	3	2.9	2.2	8	4.2	0.1	7.1	5.6
After:	8	1	5	4	3	7	10	9	6	2

Schedule List

Fig.2 An example of R-K method

[Downloaded from mail.ijaor.ir on 2026-11-27]

4.3.3 Determination of the Initial and the Final Temperature

To calculate the initial temperature, first a solution is randomly generated. Then 100 neighborhood solutions are generated. Below, you can see the initial and final temperatures calculated by equations (14-15)[19].

$$T_0 = \Delta_{\min} + 0.1(\Delta_{\max} - \Delta_{\min}) \quad (14)$$

$$T_f = 0.08T_0 \quad (15)$$

In (21), $\Delta_{\min}$ and $\Delta_{\max}$ are the minimum and maximum difference between objective functions of neighborhoods respectively and they are calculated by (16) and (17):

$$\Delta_{\min} = \min_{i,j}\{f_i - f_j\} \quad (16)$$

$$\Delta_{\max} = \max_{i,j}\{f_i - f_j\} \quad (17)$$

4.3.4 Annealing Process (Decreasing Temperature)

In this process, Lundy / Mees equation (18) has been adopted[20]:

$$T_c = \frac{T_{c-1}}{1 + \beta T_{c-1}} \quad (18)$$

In this equation, T_c is the temperature in stage c and β is a constant coefficient which is calculated by the following formula (19) where EC is the number of iterations in each temperature:

$$\beta = \frac{T_0 - T_f}{EC \times T_0 \times T_f} \quad (19)$$

4.3.5 Equilibrium Condition

In each temperature, it is necessary to investigate the equilibrium conditions to decide whether it is all right to continue the annealing process in the current temperature or it is better to stop the process and continue it after decreasing the temperature.

In some of the cases reviewed in the literature, the specific number of iterations in each temperature has been used as the equilibrium condition. In this paper, this equilibrium criterion is showed with EC. This criterion is set in the section 5-1.

4.3.6 Generating Neighborhood Solution

In order to generate neighborhood solution, two, three, or four elements of the solution array are randomly selected and their values are exchanged.

[Downloaded from mail.ijaor.ir on 2026-11-27]

4.3.7 Calculating the Procedure of the Objective Function

Objective function value (OFV) must be calculated in several stages during the proposed approach. Solutions are arrays in real numbers. Transformation of such arrays into a solution is performed in two stages:

- Finding order of customer's meets
- Allocating customers to vehicles according to capacity constraints.

In the first stage, using R-K method, real number arrays must be arranged in ascending order in order to determine the sequence of customer's meets.

In the next stage, owing to the capacity constraints of the vehicles, customers should be allocated to vehicles. According to meets of customers, they are allocated to the first vehicle until the capacity of that vehicle is not full. After the assignment of some customers to the first vehicle, the next vehicle should be considered and the remaining customers must be allocated to this vehicle, again until its capacity is not full. This process continues up to the point where all the customers are allocated to vehicles. After Allocating all customers to vehicles, OFV should be calculated based on Eq (1).

4.3.8 Calculation of Force and Solution Movements

After all SA processes reach the equilibrium state, some solutions move according to Columbus rule. This movement operation is performed merely for solutions whose OFVs are worse than the average OFV of population.

In this paper, in order to calculate forces, the equations developed by Debels *et al.* [18]are utilized. They first obtained q_{ij} by equation (20) and then calculated the force of each particle on another one by (21).

$$q_{ij} = \frac{f(x^i) - f(x^j)}{f(x^{worst}) - f(x^{best})} \tag{20}$$

$$F^{ij} = (x^j - x^i) q_{ij} \tag{21}$$

Thus, if the objective function $f(x^i)$ is larger than $f(x^j)$, particle *j* attracts *i*. According to what we discussed above, a coding procedure for solution movements is shown in Algorithm2:

Algorithm 2 - Calculation of forces and solution movements

1. $F^i \leftarrow 0$
2. For j=1 to m do
3. If $x^i \neq x^j$ then
4. $q_{ij} = \frac{f(x^i) - f(x^j)}{f(x^{worst}) - f(x^{best})}$
5. $F^{ij} = (x^j - x^i) q_{ij}$
6. $x^i \leftarrow x^i + F^{ij}$

[Downloaded from mail.ijaor.ir on 2026-11-27]

7. End if
8. End.

4.3.9 Local Search - Iterated Swap Procedure (ISP)

ISP is a local search method which is newer and faster than 2-Opt, 3-Opt[21]. This method was first introduced by Ho and Ji[22]. It was designed to improve solutions in a GA framework for scheduling purposes. In this method, five solutions are generated from the main solution (parent). These five solutions are generated through the following steps:
Step 1: Two members of the main solution are randomly selected;
Step 2: The place of these two members are exchanged;
Step 3: The place of these members are exchanged with their neighborhoods so that four solutions are generated;
Step 4: The objective function value of these solutions is calculated;
Step 5: If the best solution is better than the main solution, it substitutes the main solution.

4.3.10 Final Local Search

This local search is used just for the final solution. In this local search two or there customers from two different vehicles are selected and then if exchanging these customers improve OFV with regard to its capacity constraints, this replacement will be made. Moreover, a new vehicle may be added in the final solution and then it checks whether a customer can be inserted in this vehicle with improving OFV.

5 Computational Result

In this study, in order to verify the effectiveness of the proposed algorithm, a set of problems are randomly generated and we have compared results of PSAEM (the heuristic solution) with the lower bound. The algorithms were coded using MATLAB language, executed on a computer with a 4.00 GB RAM and an Intel Core2 Duo,2.00 GHz CPU. In the following section, at first, the PSAEM parameters are adjusted and then the lower bound and the process of generating random instances and computational results are depicted.

5.1 Parameter setting

The parameters of the proposed algorithm have remarkable effect on the quality and effectiveness of the algorithm. The PSAEM framework contains three parameters, namely equilibrium criteria (EC), generation range (GR) and population size (PS). These parameters with the help of the initial experiments, are set as follows: equilibrium criteria (EC) is taken as constant and set to 20; the generation range (GR) is also set to [-10, 10]. The population size is 30.

[Downloaded from mail.ijaor.ir on 2026-11-27]

5.2 Lower bound(Column Generation Approach)

In this paper a column generation approach is employed to obtain the lower bounds of problems. Dantzing-Wolfe decomposition of VRPDTC results in the set partitioning master problem and an Elementary Shortest Path Problem with Resource Constraints and Delivery Time Cost (ESPPRCDTC) as its subproblem. Dror [23] proved that the ESPPRC* is NP-hard in the strong sense. As the ESPPRCDTC is more general and can be reduced to the ESPPRC, it is also a NP-hard problem.

Using Dantzing-Wolfe decomposition, the master problem (M P), which consists of selecting a set of feasible paths of minimum cost, is described in equations (22)-(24), where P is the set of all feasible paths:

$$Min \sum_{p \in P} c_p \theta_p \tag{22}$$

Subject to

$$\sum_{p \in P} \alpha_{ip} \theta_p = 1, \forall v_i \in V \setminus \{v_0\} \tag{23}$$

$$\theta_p = \{0,1\}, \forall p \in P \tag{24}$$

θ_p indicates whether path $p \in P$ is selected($\theta_p = 1$) or not($\theta_p = 0$). c_p represents the cost of path p and the number of times path p meets customer v_i is determined by α_{ip}. Moreover, this model can add the fleet size constraint (25):

$$\sum_{p \in P} \theta_p \le K \tag{25}$$

As the size of the set P grows exponentially with the number of customers, we consider $MP(P_1)$ as the restriction of the M P to a subset of variables $p_1 \subset P$ and this problem is called the Restricted Master Problem. The column generation algorithm based on[24], is described in Algorithm 3.

Algorithm 3 Column generation algorithm

Generate an initial set of columns P_1
Do
  Solve $MP(P_1)$
  $\Gamma \leftarrow$ column(s) provided by the subproblem
  $P_1 \leftarrow P_1 \cup \Gamma$
While $\Gamma \neq \phi$

In this algorithm, after solving $MP(P_1)$, dual values (prices) π_i associated with customer v_i (Constraints (23)) are obtained and t_{ij} must be updated based on equation (26) to solve the subproblem.

$$t_{ij} = t_{ij} - \pi_i \quad \forall v_i \in V \tag{26}$$

As mentioned before, the subproblem of this problem is ESPPRCDTC. In each iteration of column generation algorithm, a new column is obtained from solving ESPPRCDTC. In this

* Elementary Shortest Path Problem with Resource Constraints

[Downloaded from mail.ijaor.ir on 2026-11-27]

paper, we have adapted Feillet *et al*'s approach for solving ESPPRCDTC. Feillet et al. [25] proposed an exact algorithm to solve the ESPPRC. A full description of the algorithm can be found in[25]. Here we give only a brief description of this algorithm. In the Feillet *et al*'s algorithm, each path, from the source to a node in the network, is assigned a label, which is a vector of (1) the cost of that (partial) path, (2) the resources consumed so far (vehicle load and elapsed time), (3) a resource corresponding to each customer node indicating whether the customer has already been visited or not, and (4) the total number of unreachable customers. Customers are considered *unreachable* if they have already been visited in the path or if they cannot be served by the vehicle without violating a resource constraint. We considered fuel and delivery time costs as the cost of (partial) path. Dominance rules are used to compare partial paths arriving at the same location and to discard some of them. It is noteworthy that the efficiency of this approach is affected by the tightness of the constraints and in this paper, we have just capacity constraint and we do not have time window constraint.

5.3 Random instances

In order to verify the effectiveness of the proposed algorithm, some of the randomly generated instances are presented here. In these problems, the number of customers are defined in 13 classes and 10 instances are generated in each class. Hence, 130 instances are generated based on parameters that are shown in Table2.

The demand of each customer is drawn from a lognormal distribution with parameters $\sigma = 1$ and $\mu = \ln(d') - \frac{\sigma^2}{2}$ based on [26] . This type of distribution is preferred over a uniform as it allows for a small number of customers with a high demand and a low average demand. When a demand greater than the vehicle capacity is drawn, this outcome is rejected and a new value is drawn. Therefore, d' has to be greater than $\bar{d}$ in order to get an average demand of $\bar{d}$ [26].

Table 2 Parameters of random problems

Parameters	Definition	Range of parameters
$\|V\|-1$	The number of customers	n=5,10,15,16,17,18,19,20, 25,30,35,40,45
$\bar{d}$	The average demand of a customer	200
q	Capacity of each vehicle	1000
x	x coordinate of customers place	[-100,100]
y	y coordinate of customers place	[-100,100]
α	Convert Euclidean distance to time multiple	(0,1]
γ	Fuel cost of a vehicle per time unit	(0,1]
δ_i	Delivery time cost per time unit of customer i	[0,1]

In random Instances, the distance between each two nodes is calculated based on Euclidean distance. Then with a multiple α , this distance is converted into time.
The results of these problems are shown in Table3.

Table 3 Cmparison of PSAEM with CG results

[Downloaded from mail.ijaor.ir on 2026-11-27]

n	Number of optimum solution (CG)	Number of solution in which OFV of		$\frac{PSAEM-CG}{CG}100\%$	Average run time (s)	
		PSAEM=CG	PSAEM > CG		PSAEM	CG
5	10	10	0	0.00	1.75	3.72
10	10	10	0	0.00	6.03	7.68
15	10	10	0	0.00	10.17	14.65
16	10	10	0	0.00	10.19	20.34
17	10	10	0	0.00	13.49	22.43
18	10	10	0	0.00	13.65	18.86
19	10	10	0	0.00	13.84	24.14
20	10	10	0	0.00	15.94	23.01
25	2	1	9	0.63	28.2	31.15
30	0	0	10	0.85	38.53	34.46
35	0	0	10	1.61	37.48	41.92
40	0	0	10	2.19	50.11	101.54
45	0	0	10	3.11	59.91	241.51

In Table 3, column 1 and 2 give the number of customers n and the number of instances for which an integral (optimal) solution was identified by CG. Column 3 compares objective function values of these two algorithms. Column 4 reports the average percentage gap between objective function values of these two algorithms, which is calculated as: $\frac{PSAEM-CG}{CG}100\%$. The last column compares the average runtime (s) of these two algorithms.

As we can observe in Table 3, PSAEM and CG algorithm has obtained the optimum solution in the smaller instances with at most 20 customers. For the larger instances (25-45 customers) column generation has not obtained integral solutions except two instances, therefore OFV of CG is considered as lower bound. In these instances, average deviation of PSAEM from CG is very small and on average, the run time of the PSAEM is less than CG. These comparisons illustrate the efficiency of the proposed hybrid algorithm.

Table 4 The results of the computations with 30 customers and different values of $\bar{d}$

No.	$\bar{d}$	$\frac{PSAEM-CG}{CG}100\%$	Average run time (s)	
			PSAEM	CG
1	50	8.35	31.81	49.39
2	100	6.92	29.38	48.96
3	120	3.05	28.24	41.99
4	150	2.09	29.63	51.79
5	180	1.42	33.52	34.22
6	200	0.85	38.53	34.46
7	250	0.31	27.41	30.91
8	300	0.17	29.77	28.15

We also investigate the impact of different values of average demand ($\bar{d}$) on the computational efficiency of PSAEM algorithm. The average demand influences the number of customers that can be delivered by one vehicle. The results of the computations with different values of $\bar{d}$ are presented in Table 4. In each case, we generated 10 instances and the values of n and q are fixed on 30 and 1000. As we can observe in Table 4, when $\bar{d}$ increases, the average deviation of PSAEM from CG decreases and the efficiency of the

[Downloaded from mail.ijaor.ir on 2026-11-27]

proposed hybrid algorithm increases. We can also conclude that the different values of average demand do not have considerable effect on run time of PSAEM algorithm.

6 Conclusions

This paper studied the vehicle routing problem with delivery time cost and proposed a hybrid metaheuristic approach based on Electromagnetism and simulated annealing algorithms. In the metaheuristic algorithm, the EM algorithm is utilized along with the parallel simulated annealing. In the proposed algorithm, electromagnetism creates an excellent variation in the population and Parallel Simulated Annealing is powerful in escaping from local optimums. For the assessment of the proposed algorithm, we compared it with the column generation in some random instances and it obtained the optimal and near-optimum solution in the majority of instances. For future studies, it is suggested to work on other delivery time cost function such as general piecewise cost function and exact algorithm for solve these models.

References

1. Deflorio, F. P., Perboli, G., Tadei, R., (2010). Freight distribution performance indicators for service quality planning in large transportation networks. Flex Serv Manuf J 22: 36-60.
2. TAN, K. C., CHEW, Y. H., LEE, L. H., (2006). A Hybrid Multiobjective Evolutionary Algorithm for Solving Vehicle Routing Problem with Time Windows. Computational Optimization and Applications 34(1): 115–151.
3. Xu, Y., Wang, L., Yang, Y., (2012). A New Variable Neighborhood Search Algorithm for the Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows. Electronic Notes in Discrete Mathematics 39: 289-296.
4. Liberatore, F., Righini, G., Salani, M., (2011). A column generation algorithm for the vehicle routing problem with soft time windows. J Oper Res 9: 49-82.
5. Figliozzi, M. A. ., (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transportation Research Part C 19 18 668–679.
6. Qureshi, A. G., Taniguchi, E., Yamadaa, T., (2010). Exact solution for the vehicle routing problem with semi soft time windows and its application. Procedia Social and Behavioral Sciences 2: 5931–5943.
7. Qureshi, A. G., Taniguchi, E., Yamada, T., (2009). An exact solution approach for vehicle routing and scheduling problems with soft time windows. Transportation Research Part E 45: 960–977.
8. Tang, J., ZhendongPan, Richard Y.K.Fung, HenryLau., (2009). "Vehicle routing problem with fuzzy time windows." Fuzzy Sets and Systems 160: 683–695.
9. Xu, J., Yan, F., Li, S., (2011). Vehicle routing optimization with soft time windows in a fuzzy random environment. Transportation Research Part E 47: 1075–1091.
10. Brito, J., Martinez, F. J., Moreno, J. A., Verdegay, J. L., (2012). Fuzzy optimization for distribution of frozen food with imprecise times. Fuzzy Optim Decis Making.
11. Favaretto, D., Moretti, E., Pellegrini, P., (2007). Ant colony system for a VRP with multiple time windows and multiple visits. Journal of Interdisciplinary Mathematics 10(2): 263-284.
12. Belhaiza, S., Hansen, P., Laporte, G., (2013). A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Computers & Operations Research.
13. Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., Yagiura, M., (2005). Effective local search algorithms for routing and scheduling problems with general time window constraint. Transportation Science 39: 206–232.
14. Ibaraki, T., Imahori, S., Nonobe, K., Sobue, K., Uno, T., Yagiura, M., (2008). An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. Discrete Applied Mathematics 156: 2050 – 2069.
15. Chang, P., Chen, S., Fan, C., (2009). A hybrid electromagnetism-like algorithm for single machine scheduling problem. Expert Systems with Applications 36: 1259–1267.

[Downloaded from mail.ijaor.ir on 2026-11-27]

16. Hiroyasu, T., Miki, M., Ogura, M., (2000). Parallel Simulated Annealing using Genetic Crossover. Proceedings of the IASTED International Conference parallel and distributed computing and systems 139-144.
17. Birbil, S. I., Fang, S. C., (2003). An Electromagnetism-like Mechanismfor Global Optimization. Journal of Global Optimization, 25: 263-282.
18. Debels, D., Reyck, B. D., Leus, R., Vanhoucke, M., (2006). A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. European Journal of Operational Research, 169: 638–653.
19. Connolly, D. T., (1990). An improved annealing scheme for the QAP. European J. of Operational Research, 46: 93–100.
20. Lundy, M., Mees, A., (1986). Convergence of an annealing algorithm. Mathematical Programming, 34: 111–124.
21. William, H., George, T. S. H., Ping, J., Henry, C. W. L., (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. Engineering Applications of Artificial Intelligence 21: 548–557.
22. Ho, W., Ji, P., (2003). Component scheduling for chip shooter machines: A hybrid genetic algorithm approach. Computers and Operations Research, 30: 2175–2189.
23. Dror, M., (1994). Note on the complexity of the shortest path models for column generation in vrptw. Operations Research, 42(5): 977–978.
24. Feillet, D., (2010). A tutorial on column generation and branch and price for vehicle routing problem. Q J Oper Res 8: 407-424.
25. Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., (2004). An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. Networks 44(3): 216–229.
26. Jong, C. D., Kant, G., Vliet, A. V., (1998). On finding minimal route duration in the vehicle routing problem with multiple time windows. Department of Computer Science, Utrecht University.

[Downloaded from mail.ijaor.ir on 2026-11-27]