Journal of AI and Data Mining
Vol 7, No 1, 2019, 169-179

DOI: 10.22044/JADM.2018.7116.1840

A Combined Metaheuristic Algorithm for the Vehicle Routing Problem and its Open Version

M. Yousefikhoshbakht [1] and N. Mahmoodi Darani [*2]

1. Department of Mathematics, Faculty of Sciences, Bu-Ali Sina University, Hamedan, Iran
2. Hashtgerd Branch, Islamic Azad University, Tehran, Iran.

Received 27 May 2018; Revised 31 August 2018; Accepted 14 November 2018

**Corresponding author: khoshbakht@basu.ac.ir (M.Yousefikhoshbakht).*

Abstract

The Open Vehicle Routing Problem (OVRP) is one of the most important extensions of the Vehicle Routing Problem (VRP) that has many applications in the industry and services. In VRP, a set of customers with a specified demand of goods is given, where a fleet of identical capacitated vehicles is located. The ‘‘traveling costs’’ between the depot and all the customers, and between each pair of customers are also defined. In OVRP against VRP, the vehicles are not required to return to the depot after completing service. Since VRP and OVRP belong to the NP-hard problems, in this work, an efficient hybrid elite ant system called EACO is proposed for solving them. In this algorithm, a modified tabu search, a new state transition rule, and a modified pheromone-updating rule are used for more improved solutions. As a result of these modifications, the proposed algorithm is not trapped at the local optimum and discovers different parts of the solution space. The computational results of 14 standard benchmark instances for VRP and OVRP show that EACO finds the best known solutions for most of the instances, and it is comparable in terms of solution quality to the best performing published metaheuristics in the literature.

Keywords: *Vehicle Routing Problem, Open Vehicle Routing Problem, Elite Ant System, Tabu Search, NP-Hard Problems.*

1. Introduction

The vehicle routing problem (VRP) is one of the most famous problems in combinatorial optimization problems analyzing efficient routes with the minimum total cost for a fleet of vehicles for serving some commodity to a given number of customers. Each customer is visited exactly once by one vehicle, while the vehicle activity is bounded by capacity constraints, duration constraints, and time constraints. Each route is a sequence of customers that starts at the depot and finishes at one of the customers or each route is a sequence of customers that begins at a defined customer and ends at the distribution depot, where goods are gathered [1]. From a graph theoretical viewpoint, VRP is defined as a complete undirected graph G = (V,E), in which V = {0, . . . , n} and $E = \{(i, j) : i, j \in V, i \neq j\}$. Vertex 0 represents the depot and the other vertices represent the customers. The cost of travel from vertex i to vertex j is denoted by c_{ij}. A fleet of K identical vehicles located at the depot, each of capacity $Q > 0$, is given and each customer i has a non-negative demand $0 < q_i < Q$. Each customer must be serviced by only one vehicle, and no vehicle may serve a set of customers whose total demand exceeds its capacity. The objective is to define the set of vehicle routes that minimizes the total costs such that each vehicle starts at the depot and end at it after visiting some customers. The description of this important variant of VRP appeared in the literature over 30 years ago but has just recently attracted the attention of scientists and researchers [2]. If the sum of demands of each cycle in figure 1 is less than Q, this is a feasible solution for VRP.

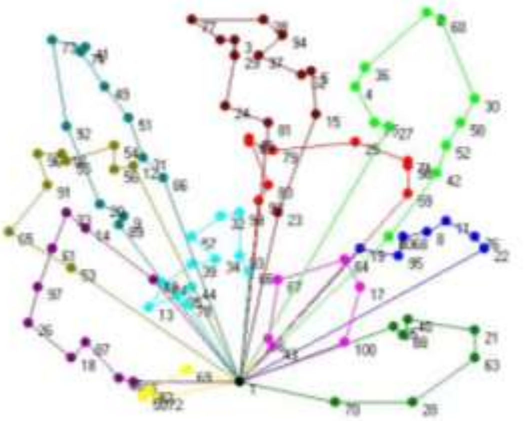

Figure 1. A feasible solution for VRP.

The open vehicle routing problem (OVRP) differs from the well-known VRP in that the vehicles do not necessarily return to the depot after delivering goods to customers and ends its route in a customer. In the recent years, OVRP has been envisaged in many practices such as the home delivery of packages and distribute newspapers. Furthermore, companies that use contractors to deliver newspapers to residential customers do not require the contractors and their vehicles to return to the depot. These real life applications of the OVRP concerns the case, where the company does not have vehicles at all or the vehicles owned by the company are not enough to use them for the distribution of the products between the customers. In both cases, the company has to hire some vehicles to realize the distribution of the products. When the vehicles finish their jobs, they do not return to the depot. This problem also belongs to the category of the third party logistics (3PL) problems. As a result, the researcher's interest in OVRP has increased dramatically, and a wide variety of new algorithms has been developed to solve the problem over the last twenty years. This problem, similar to VRP, involves routing a homogeneous fleet of vehicles with fixed capacity Q that start to move simultaneously from the depot but not come back to the depot after visiting the customers. In other words, each route in OVRP is a Hamiltonian path and maybe a route-length constraint to limit the maximum distance traveled by each vehicle. Each customer has a known demand and is serviced by exactly one vehicle. The objective is to design a set of minimum cost routes to serve all customers [3]. In addition, we need to find the minimum number of vehicles required to deliver goods to all customers. Figure 2 shows a feasible solution for OVRP with 10 vehicles and 20 customers.

From the combinatorial optimization viewpoint, the main difference between VRP and OVRP is that in the first case, the route is a Hamiltonian cycle, while in the second case, the route is a Hamiltonian path [4]. On the other hand, OVRP turns out to be more common than VRP in the sense that any closed version with *n* customers can be converted into an open version of VRP with *n* customers but the transformation in the reverse direction is not possible. Figure 3 shows the feasible solutions to both the open and closed versions of VRP for the same input data. In general, this figure shows that the feasible solution for the open version of VRP can be different from that for the closed version. In this figure, the depot and customer are represented by square and circle, respectively.

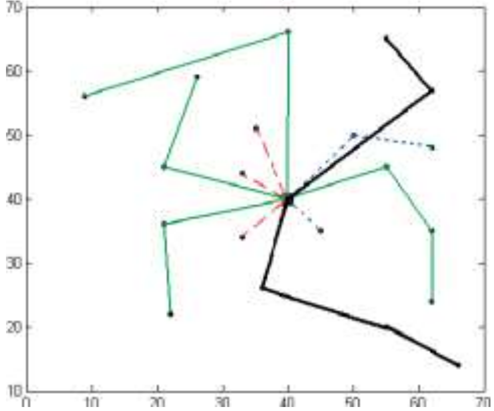

Figure 2. A feasible solution for OVRP.

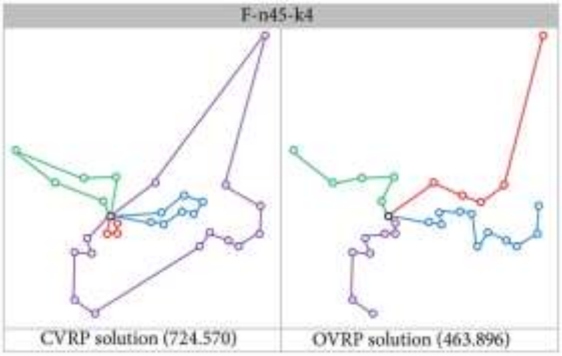

Figure 3. Two solutions for VRP and OVRP.

OVRP has received sparse attention in the literature compared to VRP [5]. While the earliest description of OVRP offered by Schrage [6] appeared in the literature over 20 years ago, OVRPs have just recently attracted the attention of practitioners and researchers. He dedicated to the description of realistic routing problems.

As it has been mentioned, OVRP consists of Hamiltonian paths originating at the depot and terminating at one of the customers, and the Hamiltonian path problem is equivalent to the traveling salesperson problem, which is known to be NP-hard [7], then the best Hamiltonian path is NP-hard. Besides, this problem with a fixed source node must be solved for each vehicle in OVRP, and the OVRP solutions involve finding the best Hamiltonian path for each set of customers assigned to a vehicle. Consequently, OVRP is also an NP-hard problem. For this reason, exact algorithms cannot solve most of the practical examples of this problem to optimality within a reasonable time and the algorithms used in practice are the heuristic and metaheuristic algorithms. These approaches can find the optimal or near optimal solutions within a reasonable computing time. For example, a tabu search (TS) algorithm has been proposed by Fu et al. [8], in which the initial solution is provided by a ‘furthest first heuristic’ and the exchanges are based upon the two-interchange generation mechanism. In this algorithm, a combination of vertex reassignment, 2-opt, vertex swap, and ‘tails’ swap within the same route or between two routes are used simultaneously.

Ozyurt et al. have presented a modified Clarke-Wright parallel savings algorithm, the nearest insertion algorithm, and a tabu search heuristic for the open vehicle routing problem with time deadlines [9]. Some random test problems and a real-life school bus routing problem have been considered and solved by these heuristics. Finally, the results of this algorithm have been compared with other algorithms.

Li et al. [10] have developed a variant of record-to-record travel algorithm for the standard OVRP that avoids the premature convergence and found high quality solutions in a short computing time. In this algorithm, a fixed-length neighbor list with 20 customers is used, and they generate an initial feasible solution using a sweep algorithm. Besides, the minimum number of vehicles required to service all the customers is calculated. They use each customer as a starting point in the sweep algorithm so that one solution is generated for each customer. If no solution uses the minimum number of vehicles, the solution is selected that uses the smallest number of vehicles to service all the customers.

Repoussis et al. have considered OVRP with time windows (OVRPTW) in which customers' service can take place within fixed time intervals that represent the earliest and latest times during the day [11]. They formulate a comprehensive mathematical model to capture all aspects of the problem, and incorporated all the critical practical concerns. The model is solved using a greedy look-ahead route construction heuristic algorithm, which utilizes time windows related information via composite customer selection and route-insertion criteria. The computational results on a set of benchmark problems from the literature provide very good results, indicating the applicability of the methodology in real-life routing applications.

Pisinger and Ropke [12] also have offered an effective metaheuristic based on adaptive large neighborhood algorithm, in which customers are removed randomly from the current position and reinserted in the place with the cheapest possible route. Furthermore, for diversifying and intensifying the search, some removal and insertion heuristics are used. Moreover, several well-known metaheuristics have been proposed for the versions of OVRP involving only capacity constraints. For example, in 2005, Tarantilis et al. offered a population-based algorithm and a heuristic based on the threshold-accepting type for solving OVRP [13].

Marinakis et al. have presented a relatively new swarm intelligence algorithm called BBMO that simulates the mating behavior for solving OVRP [14]. For testing the quality of the algorithm, two sets of instances were considered, and the results obtained showed that the proposed algorithm was very satisfactory in most instances.

A real-world problem has been proposed by an international company in Spain and modeled as a variant of OVRP by López-Sánchez et al. [15]. In this problem, the maximum time spent on the vehicle by one person must be minimized. Thus, a metaheuristic algorithm was proposed to obtain high quality solutions. In order to analyze the algorithm, 19 school-bus routing problems in the literature were considered on nine hard real-world instances.

Brito et al. have proposed the close–open vehicle routing problem, where the routes can be opened and closed [16]. This variant is nowadays a standard practice model in business. Furthermore, they formulated a model of this novel variant with time windows, and a hybrid metaheuristic was proposed for its solutions. This algorithm was applied to a real problem with outsourcing. Finally, Erbao et al. have proposed OVRP with uncertain demands. In this paper, firstly, the customer's demand was described and then an optimization model was proposed to minimize the transportation costs. They have also proposed four strategies to handle the uncertain demand and an improved evolution algorithm to solve the robust model. Furthermore, the performance of four different robust strategies was analyzed by considering the extra costs and unmet demand.

In [17], a variable neighborhood search-based algorithm has been proposed to solve the newspaper delivery optimization problem for a media delivery company in Turkey by reducing the total cost of carriers as a real-world OVRP problem. The results of the proposed algorithm on varieties of small- and large-scale benchmark suites show that not only the algorithm provides either the best known solution or a competitive solution for each benchmark instance but also the real-world company's solutions is also improved by more than 10%.

Finally, Niu et al. studied fuel consumption in the context of third party logistics, and the mathematical model of the green open vehicle routing problem with time windows (GOVRPTW) was described based on the comprehensive modal emission model (CMEM) in their work [18]. Furthermore, they proposed a hybrid tabu search algorithm involving several neighborhood search strategies to solve this problem. Computational experiments were performed on realistic instances based on the real road conditions of Beijing, China. The effect of empty kilometers was analyzed by comparing different cost components. Compared with the closed routes, the open routes reduced the total cost by 20% with both the fuel emissions costs and the CO_2 emissions cost down by nearly 30%. For the experiments with congested nodes, the fuel and emissions cost rose by 12.3%, and the driver cost even increased by 31.3%.

In the last years, some publications using different exact, heuristic, and metaheuristic algorithms for VRP and OVRP have been published. Since these problems are NP-hard problems, the instances with a large number of customers cannot be solved in optimality within a reasonable time period. For this reason, a large number of approximation techniques have been proposed for its solution in the recent ten (V,E) is given (if the graph is not complete, we can consider the lack of each arc with a infinite size) with years. These techniques have been classified into three main categories including the classical heuristics, the single solution-based metaheuristics, and the population-based metaheuristics. Besides, according to some shortcomings like its slow computing speed and local-convergence in ACO, the basic of this algorithm cannot be directly applied to the problem with an acceptable performance, and few researchers have proposed new methods to improve the original ACO and applied them. Therefore, to achieve the effectiveness and efficiency of ACO, we try to improve the quest for the performance of hybrid algorithms. As a result, in this work, an efficient hybrid elite ant system with tabu search called EACO is proposed to improve both the performance of the algorithm and the quality of the solutions. The proposed algorithm uses the elite ant system (EAS) for solving the VRP and OVRP problems and then improves the global ability of the algorithm. When the best-found solution is not changed for five times in EAS, the tabu search has been used as an effective local search for *n* best solutions until now (n is the number of customers for that instance). Then if the quality of the best solution of tabu search is not increased for five times, these solutions are considered and released with pheromone. These steps are continued until the stop condition is satisfied. The results in the fourteen instances proposed by Christofides and the two problems represented as F11 and F12 by Fisher show that the proposed algorithm can obtain high quality solutions compared to the other metaheuristic algorithms.

The structure of the remainder of the paper has been organized as what follows. In the next sections, the model of the problem and the proposed EACO are explained, respectively. This algorithm mainly consists of the iteration of the three steps including *n* ants build the solution independently, apply the tabu search (TS) algorithm to improve the solution, and update the global pheromone information. In this section, we describe each step in more details. In Section 4, the proposed algorithm is compared with some of the other algorithms on standard problems belonging to the VRP and OVRP library. Some concluding remarks are given in the final section.

2. Description and formulation

From a graph theoretical viewpoint, we can define OVRP as follows. A complete undirected graph G = V = {0, . . . , n} and $E = \{(i, j) : i, j \in V, i \neq j\}$. Vertex 0 represents the depot, and the other vertices represent customers. The cost of travel from vertex i to vertex j is denoted by c_{ij}. A fleet of K identical vehicles located at the depot, and each capacity $Q > 0$, is given. Each customer i has a non-negative demand q_i with $0 < q_i < Q$. Each customer must be serviced by a single vehicle, and no vehicle may serve a set of customers whose total demands exceed its capacity. Each vehicle route must start at the depot and end at the last customer it serves. The objective is to define the set of vehicle routes that minimizes the total costs such that each vehicle starts at the depot and ends at a customer. We present the following mathematical formulation for OVRP using variables and y_{ij}, where x_{ij} takes the value of one if a vehicle travels directly from customer *i* to customer *j*, and 0 otherwise denotes the route. The flow variables y_{ij} specify the number of goods that a vehicle is carrying when leaves customer *i* to service customer *j*.

$$\text{Min}\sum_{k=1}^{K}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}^{k} x_{ij}^{k} \tag{1}$$

subject to

$$\sum_{k=1}^{K}\sum_{i=0}^{n} x_{ij}^{k} = 1 \qquad \forall j = 1,2,...,n \tag{2}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{n} x_{ij}^{k} \le 1 \qquad \forall i = 1,2,...,n \tag{3}$$

$$0 \le \sum_{i=1}^{n} x_{ij}^{k} - \sum_{i=1}^{n} x_{ji}^{k} \le 1 \quad \forall j(k) = 1,2,...,n(K) \tag{4}$$

$$\sum_{k=1}^{K}\sum_{i=0}^{n} y_{ij}^{k} - \sum_{k=1}^{K}\sum_{i=0}^{n} y_{ji}^{k} = q_j \quad \forall j = 1,2,...,n \tag{5}$$

$$q_j x_{ij}^{k} \le y_{ij}^{k} \le (Q - q_i) x_{ij}^{k} \quad \forall i,j,k = 0,1,...,n(K) \tag{6}$$

$$\sum_{j=1}^{n} x_{i0}^{k} = 0 \qquad \forall k = 1,2,...,K \tag{7}$$

$$x_{ij}^{k} \in \{0,1\} \qquad \forall i,j = 0,1,...,n, i \ne j, \forall k = 1,2,...,K \tag{8}$$

$$y_{ij}^{k} \ge 0 \quad \forall i,j = 0,1,...,n, \ \forall k = 1,2,...,K \tag{9}$$

The objective function (1) gives the sum of the total variable routing cost. Constraints (2) mean that only one arc can be entered for each customer; however, constraints (3) show that almost one arc can be exited from each customer. Constraints (4) state that if a $\kappa_{ij} = c_{i0} + c_{0j} - c_{ij}$, where node 0 is the depot and vehicle visits a customer, it can remain there or depart from it. Equality equations (5) proved that the demands of all customers were fully satisfied. Constraints (6) state that the vehicle capacity is never exceeded. Constraints (7) guarantee that there is no arc from each customer to the depot. Constraints (8) describe that each arc in the network has the value one if it is used, and 0 otherwise. Finally, restrictions (9) force the flow to remain non-negative.

3. Proposed algorithm

The ant colony optimization (ACO) was inspired by the behavior of real ant colonies in nature in order to find routes between their nests and food sources. In 1991, Dorigo et al. used this concept and proposed ACO to solve the combinational optimization problems. In this section, an efficient hybrid EACO is proposed to solve VRP and OVRP, in which the best solutions constructed by ants until now are ranked in each iteration. Then TS is used as an improved procedure. EACO has made three main contributions, as follow:

3.1. Building solution

The first phase of EACO is solution construction, in which for *n* groups, *m* ants are initially positioned on the depot and each ant of the colony efforts to build a feasible solution represented as a single route based on the pheromone trail and heuristic information. The proposed EAS presents a new transition rule to find the better customer for each vehicle in every iteration. According to the following transition rule in formula (10), the next node j from node i in the route is selected by ant k among the unvisited nodes J_i^k,

$$P_{ij}^{k}(t) = \frac{\tau_{ij}^{\alpha}(t)\kappa_{ij}^{\beta}(t)}{\sum_{r\in J_i^k}\tau_{ir}^{\alpha}(t)\kappa_{ij}^{\beta}(t)} \qquad \forall j \in j_i^k, \tag{10}$$

where

$\tau_{ij}(t)$: The amount of pheromone on edge joining nodes i and j.

$\kappa_{ij}(t)$: The savings of combining two nodes on one tour as opposed to serving them on two different tours. The savings of combining any two customers i and j are computed as

c_{ij} denotes the distance between nodes *i* and *j*.

α *and* β : The control parameters.

3.2. Applying TS

The literature on algorithms tells us that in order to find high-quality solutions by metaheuristics, a powerful local search algorithm is required [19-22]. Therefore, among the best solutions proposed, TS is used as a local search algorithm until the best obtained solution is not improved for five iterations. In figure 4, a diagram for the proposed algorithm is shown; the second figure presents the solution obtained by EAS and the third one shows the solution after using TS.

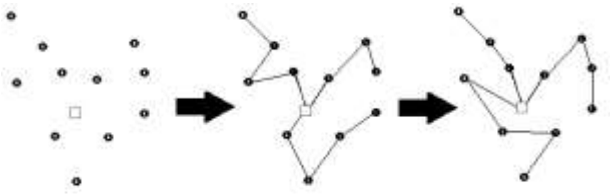

Figure 4. General diagram of EACO.

The TS algorithm is one of the most important local search algorithms that can obtain high quality solutions for many optimization problems. This algorithm requires an initial solution from which the search process begins, namely the definition of the neighborhood and the consequent first move. The pseudo-code of TS is shown in figure 5 [23]. In applying TS to VRP and OVRP, most of the studies have paid little attention to the initial solution. Generally, the initial solution, i.e. assigning each customer to a route, is trivial or is obtained with a fast and well-known heuristic. We think that the main reasons for this are the belief that the initial solution has very little influence on the quality of the final solution, and the need for finding a starting solution very quickly, leaving the improved work for the TS algorithm. This practice has not impeded the attainment of very good results because TS is effective when correctly applied. However, in this algorithm, the initial solutions are calculated with EAS and still give an important contribution to enhance the final solution. In the selection of the method to produce the initial solution, our goal was to find a solution with a good structure, and less importance was given to its cost. Thus in this algorithm, n best found solutions to EAS were considered as the initial solutions to TS.

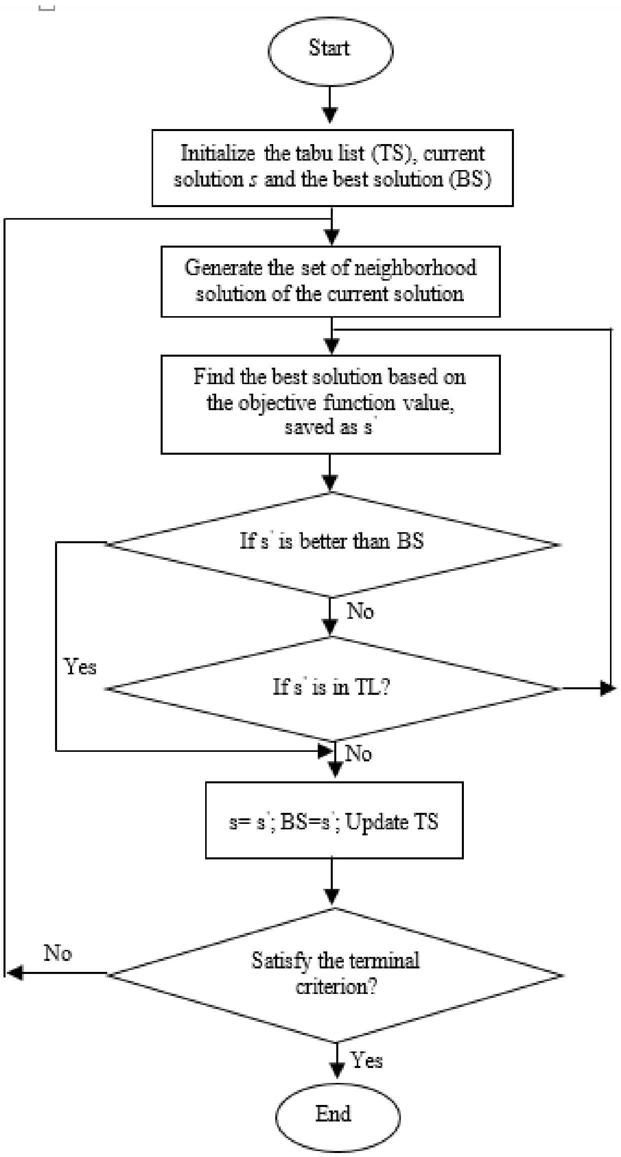
Start
Initialize the tabu list (TS), current solution *s* and the best solution (BS)
Generate the set of neighborhood solution of the current solution
Find the best solution based on the objective function value, saved as s'
If s' is better than BS
No
Yes
If s' is in TL?
No
s= s'; BS=s'; Update TS
No
Satisfy the terminal criterion?
Yes
End

Figure 5. Pseudo-code of TS.

The proposed TS comprises three types of neighborhood moves including the 2-Opt, insert, and swap moves. Although all customers are the candidates to be moved, n number of neighborhoods are produced by the mentioned algorithms, in which 30, 35, and 35 percent of them belong to the 2-Opt, insert, and swap exchanges, respectively. It is to be noted that these moves are not equally performed in each iteration because of diversifying the search and keeping the computing time at reasonable levels. In multiple routes, edges (i,i+1) and (j,j+1) that form a criss-cross and belong to different routes are considered, and the 2-Opt move is applied. The insert move transfers a node from its position in one route to another position in a different one. In the swap move, two nodes from different routes are selected and changed. The same procedure is conducted in the case of multiple routes.

It is to be noted that in this step, tabu list (TS) is used to prevent the return to the most recently visited solutions for a specific number of iterations (tabu tenure) in order to avoid cycling. At this time, "aspiration criteria" are used for some of the tabu solutions, which must now be avoided, could be of excellent quality, and might not have been visited. In the proposed algorithm, it is possible to move from the current solution to the best solution in its neighborhood that should not be in the TL or satisfies some aspiration criteria. For a strong diversification technique in the proposed algorithm, the size of TL is considered as a variable. In more details, if MTS cannot improve the best known solution for a pre-specified number of iterations, direction of the proposed algorithm should change towards a part of solution space that has not been explored yet. Therefore, the length of TL is increased. After the diversification policy, the search process is increased by declining the value of TL for a number of consecutive iterations. At this stage, if TS cannot improve the solution for five iterations, all the n solutions are returned to EAS.

2.3. Global pheromone updating

The pheromone updating of EAS includes the local and global updating rules. In contrast to AS, the pheromone of all edges belonging to the route obtained by ants called local updating will not be used in EAS. In addition, EAS uses only global updating after producing the solution to VRP or OVRP in the current iteration. In other words, when quality of the best solution until now is not increased for five times, the proposed modified TS is used to improve it. After applying this algorithm for n best known solutions, the global updating is applied.

In this step, the arcs belonging to the n solutions are released with pheromone and are encouraged with the constant coefficient *e* based on Formula (11). This process causes that the arcs belonging to the best routes until now in any iteration are more highlighted, and to be updated according to the value of the route L^f for solution f. It is to be noted that the less the value for L^f the more pheromones are released on the arcs. In the proposed algorithm, when the best solution until now is not changed for five times, the modified TS algorithm is used to improve the new solution.

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}^{\ f}(t), \qquad (11)$$

where:

f: Number of the best solutions.

ρ: A parameter in the range [0, 1] that regulates the reduction of pheromone on the edges.

T^f: The collection of arcs passed over by the ant f with the best solution until now.

$$\Delta\tau_{ij}^{f}(t) = \begin{cases} e / L^f(t) & (i,j) \in T^f \\ 0 & (i,j) \notin T^f \end{cases}, \qquad (12)$$

e: A constant coefficient determined by the ser.

At this stage, if the best found solution until now is iterated for 20 times, the algorithm ends and the results obtained and values up to now are considered as the best values and results of the algorithm. Otherwise, the algorithm is iterated by returning to the transition rule step. Figure 6 shows the pseudo-code of the proposed algorithm.

4. Computational experiments

In this section, the results of the proposed algorithm are compared with other algorithms for solving the VRP and OVRP instances. Since EACO is a meta-heuristic algorithm, the best solution found for ten independent runs is reported in the next tables for the VRP and OVRP instances. The algorithm is implemented in *C* programming language and runs on a 3.5 GHz Intel Pentium 3 processor and 4 GB of RAM running Microsoft Windows 7 Ultimate. There are 14 test problems denoted as C1–C14, taken from Christofides et al. in 1979 [24] and identified by their original number, prefixed, respectively, with the letters C available in the literature. The cost of an edge is then taken to be equal to the Euclidean distance and computed with real numbers. We had to decide the precision of computation in these distances. Besides, these benchmark instances with 50–199 customers have been widely used as benchmarks. The first ten instances have customers that are randomly distributed around the depot. In the last four instances, the customers appear in clusters, and the depot is not centered. All the test instances are subjected to capacity constraints, while problems 6–10, 13, and 14 also have the route length limitations. The information for the 14 instances is shown in table 1.

```
Initialize pheromone trails, alpha, beta, e, n, k=0,
  $s^* = \phi(\text{The best solution}), v^* = +\infty(\text{The best value})$.
  While (k<=10)
  Begin
      Construct n solutions $s_i$ by using formula (1)
      Rank the solutions and select the f best solutions until now.
      If $v_{current} < v^*$ ($s^*=s_{current}$;k=0),else k=k+1;k'=0;
      If (k==5)
          {While (1)
              Apply TS for f best solutions;
              If $v_{current} < v^*$ ($s^*=s_{current}$;k'=0),else k'=k'+1;
              If (k'==5) Break;
          End}
      Global update pheromone for the BKSs.
  End
  Show $s^*$ and $v^*$.
    End // procedure //
```

Figure 6. Pseudo-code of EACO

In this table, column instance shows the name of problems, and columns *n* and *k* are the numbers of customers and vehicles, respectively, and column BKS is the best known solutions obtained by other algorithms. Besides, columns 5-8 show the results of genetic algorithm (GA) [25], scatter search algorithm combined by ant colony optimization (SS_ACO) [26], particle swarm intelligent (PSO) [27], and genetic algorithm and particle swarm intelligent (GAPSO) [28]. GA is the weakest algorithm among all the presented algorithms in table 1 because it is only able to find the best solutions in one of the fourteen examples. In comparison with GA, SS_ACO has been able to find better solutions and come up with the best solution in 12 examples. PSO is another metaheuristic that has failed to improve the solutions in 10 examples and has come up with solutions similar to the ones found by GA. From the comparison between GAPSO and the proposed algorithm, it can be seen that GAPSO in the two examples has been able to find better solutions than the proposed algorithm. However, EACO also has found better solutions than this algorithm for two examples. Generally, the results of this table show that EACO can find the optimal solution for 11 out of 14 problems, and is a competitive algorithm compared to BKS. Furthermore, the gap between other problems is less than 1% and so the proposed algorithm finds nearly the best known solutions.

In addition to the VRP problems, the results of the proposed algorithm are compared with other algorithms for OVRP instances. Thus the fourteen problems denoted as C1-C14 and two problems represented as F11 and F12 taken from Fisher are considered in table 2. In this table, the first column gives the instance name, and the second till seventh column show the results of six algorithms. Finally, to show the EACO performance more clearly, we present BKS, published in the related literature. It is to be noted that some algorithms use a different number of vehicles in this table. Besides, each algorithm consists of two sub-column including the best gained solution and CPU time. All the CPU times reported in the tables are in seconds. We compare the results obtained by the proposed algorithm on the above-mentioned instances with some algorithms including TSF and TSR based on TS by Fu et al. [29], TSAN based on TS by Brandao [4], ORTR used record-to-record travel algorithm to handle very large instances of the standard OVRP by Li et al. [30], ALNS with 50,000 iterations based on adaptive large neighborhood search used the minimum spanning tree by Pisinger and Ropke [12], VNS based on a variable neighborhood search by Fleszar [31].

The results obtained show that with the minimum number of vehicles as specified by the lower bound of *K*, EMEAS finds 10 optimal solutions published in the literature and obtains nearly BKS for instances C2, C3, C5, C8, C10, and C14, and the maximum relative error is 1.05% for the instance C5 and the average relative error is 0.27%. It is to be noted that for each problem in the table, the proposed algorithm is used. Besides, the best algorithm except EACO is VNS because it finds the optimal solution for 5 out of 16 problem instances in the literature. Besides, TSF, ALNS 50K, ORTR, and TSAN can find 4, 4, 4, and 0 optimal solutions thorough these instances. These results indicate that EACO is much better than the results of these algorithms. For better comparisons between our algorithm and other algorithms for OVRP, mean gap is used and computed for all instances.

The gap is computed by Formula (4), in which a zero value indicates that the best known solution of instance in the literature is equal to the best solution found by the algorithm (BKS1).

Gap= ((BKS1-BKS)/BKS)*100 (14)

Based on table 2, the average gap for 16 instances is 2.27%, 6.38%, 1.29%, 1.37%, and 1.58% for TSF, TSAN, ORTR, ALNS, and VNS, respectively. These results show the efficiency of the proposed algorithm with 0.27% compared to the mentioned five algorithms. Moreover, ORTR performs better than ALNS, and ALNS obtains much better solution than VNS. Therefore, the algorithms in terms of their performance of mean gap from the worst to the best are: TSAN, TSF, VNS, ALNS, ORTR, and

EACO. In addition, two of the solutions found in the examples in table 2 are presented in figure 6. It should be noted that in both examples, C12 (left figure) and F12 (right figure) presented in this figure, the proposed algorithm was able to find BKS.

Table 1. Comparing Results of EACO with other metaheuristic algorithms.

Instance	n	k	L	GA	SS_ACO	PSO	GAPSO	EACO	BKS
C1	50	5	-	524.61	524.61	524.61	524.61	524.61	524.61
C2	75	10	-	849.77	835.26	844.42	835.26	835.26	835.26
C3	100	8	-	840.72	830.14	829.40	826.14	826.14	826.14
C4	150	12	-	1055.85	1038.20	1048.89	1028.42	1069.23	1028.42
C5	199	17	-	1378.73	1307.18	1323.89	1294.21	1291.45	1291.45
C6	50	6	180	560.29	559.12	555.43	555.43	555.43	555.43
C7	75	11	144	914.13	912.68	917.68	909.68	909.68	909.68
C8	100	9	207	872.82	869.34	867.01	865.94	865.94	865.94
C9	150	14	180	1193.05	1179.4	1181.14	1163.41	1193.05	1162.55
C10	199	18	180	1483.06	1410.26	1428.46	1397.51	1395.85	1395.85
C11	120	7	-	1060.24	1044.12	1051.87	1042.11	1042.11	1042.11
C12	100	10	-	877.8	824.31	819.56	819.56	819.56	819.56
C13	120	11	648	1562.25	1556.52	1546.20	1544.57	1544.57	1541.14
C14	100	11	936	872.34	870.26	866.37	866.37	866.37	866.37

Table 2. Results of EACO compared to other metaheuristic algorithms.

Instance	TSF		TSAN		ORTR		ALNS 50K		VNS		EACO		BKS
	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	Cost	Time	
C1	408.5	170.2	438.2	1.7	416.06	6.2	416.06	230	416.06	17.6	408.5	3.11	408.5
C2	587.8	202.1	584.7	4.9	567.14	31.3	567.14	530	567.14	29.0	567.14	11.35	564.06
C3	644.3	719.9	643.4	12.3	639.74	39.5	641.76	1280	639.74	239.6	622.13	21.23	617
C4	734.5	1610.3	767.4	33.2	733.13	128.6	733.13	2790	733.13	585.0	733.13	32.52	733.13
C5	878.0	2060.5	1010.9	116.9	924.96	380.6	896.08	2370	905.96	292.1	879.37	55.19	870.26
C6	400.6	128.0	416.0	1.4	412.96	10.3	412.96	310	412.96	75.8	400.6	3.75	400.6
C7	565.7	292.4	581.0	3.4	568.49	32.2	583.19	330	596.47	22.3	560.4	10.12	560.4
C8	638.2	987.8	652.1	10.4	644.63	53.2	645.16	1140	644.63	587.6	644.63	15.62	638.2
C9	758.9	1635.2	827.6	25.2	756.38	195.1	757.84	1850	760.06	1094.1	752.00	35.91	752.0
C10	891.3	1922.2	946.8	100.1	876.02	363.5	875.67	2240	875.67	1252.4	876.02	43.65	875.67
C11	753.8	735.8	713.3	15.7	682.54	121.6	682.12	1410	682.12	231.6	682.12	16.18	682.12
C12	549.9	413.4	543.2	7.8	534.24	32.9	534.24	1180	534.24	163.7	534.24	26.61	534.24
C13	943.0	741.1	994.3	25.8	896.50	120.3	909.80	1160	904.04	1820.1	896.50	29.24	896.50
C14	586.8	463.2	651.9	8.1	591.87	62.9	591.87	750	591.87	389.0	591.87	15.41	586.8
F11	178.0	256.0	179.5	5.7	177.00	19.5	177.00	1040	178.09	140.2	175.00	6.23	175.0
F12	789.7	1044.8	825.9	32.7	769.66	158.2	770.17	3590	769.66	1237.5	769.66	23.41	769.66

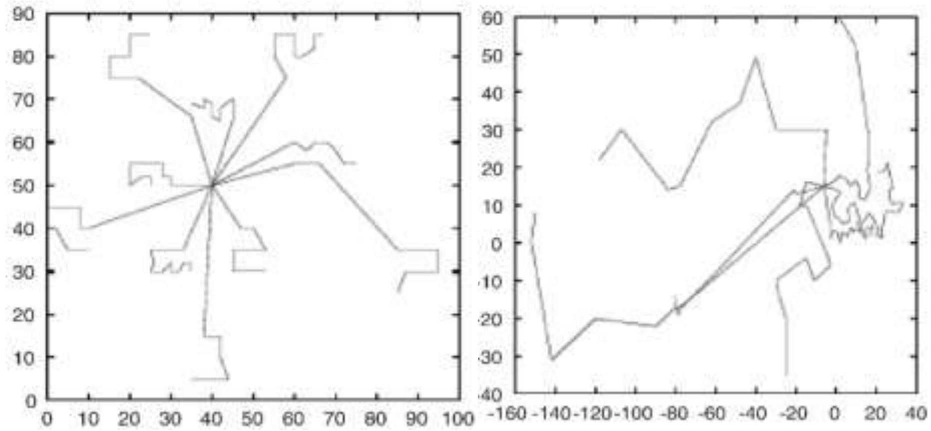

Figure 6. Some of the OVRP solutions found by EACO.

6. Conclusion

With the rapid development of the sharing economy, outsourcing logistics operations to third party logistics has become an efficient way of reducing the costs in freight transportation. It can be modeled as a variant of OVRP, where the vehicles do not return to the depot after servicing customers. This problem is different from most variants of vehicle routing problems reported in the literature, in which the vehicles do not return to the depot after delivering the last customer. The practical importance of OVRP has been established some years ago but it has received very tiny attention from scientists and researchers. In this research work, we created an effective hybrid EAS called EACO that could find very good solutions for the instances of VRP and OVRP in a short computation time. We introduced some modifications to improve the algorithm. In this algorithm, the *n* obtained best solutions until now are considered and released with pheromone. Besides, the proposed tabu search (TS) algorithm comprises three kinds of neighborhood algorithms including the 2-Opt, 0-1, and 1-1 exchanges. These moves are distinguished regarding the exchanges performed to convert one tour into another.

To improve the TS further, the size of tabu list is considered as the minimum and maximum values for the diversification and intensification policies, respectively. We compared its performance with other meta-heuristic algorithms published recently and designed for the same purpose. The results obtained showed that the proposed algorithm was efficient for both problems. For example, the average quality of gap was 0.49% and 0.27% for the VRP and OVRP instances, respectively. Besides, the average quality of gap for EACO was less than 1% for both problems when only the travel distance was minimized for the two instances proposed by Golden. The algorithm was also compared with a number of metaheuristic, evolutionary, local search, and nature inspired algorithms from the literature. The experimental results showed that the EACO algorithm was very efficient and competitive in terms of the solution quality. We are convinced that this technique will be applied in some versions of vehicle routing problems such as the vehicle routing problem with pickup and delivery or general vehicle routing problem in the future.

6. Acknowledgement

The authors would like to acknowledge the Hashtgerd Branch, Islamic Azad University for the financial support of this work.

References

[1] Atefi, R., Salari, M., Coelho, L. C. & Renaud, J. (2018). The open vehicle routing problem with decoupling points, European Journal of Operational Research., vol. 265, no. 1, pp. 316-327.

[2] Brandão, J. (2018) Iterated local search algorithm with ejection for the chain the open vehicle routing problem with time windows, Computers & Industrial Engineering, vol. 120, pp. 146-159.

[3] Bra¨ysy, O. (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows, INFORMS Journal on Computing, vol. 15, no. 4, pp. 347–368.

[4] Branda˜o, J. (2004). A tabu search algorithm for the open vehicle routing problem, European Journal of Operational Research, vol. 157, pp. 552–564.

[5] Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem, Management Science, vol. 6, no. 1, pp. 80–91.

[6] Schrage, L. (1981). Formulation and structure of more complex/ realistic routing and scheduling problems. Networks, vol. 11, pp. 229–232.

[7] Reinelt, G. (1991). The traveling salesman: Computational solutions for TSP applications. Berlin: Springer-Verlag.

[8] Fu, Z., Eglese, R. & Li, L. Y. O. (2005). A new tabu search heuristic for the open vehicle routing problem, Journal of the Operational Research Society, vol. 56, pp. 267–274.

[9] Ozyurt, Z., Aksen, D. & Aras, N. (2006). Open vehicle routing problem with time deadlines: Solution methods and an application. In Operations Research Proceedings, pp. 73-78. Springer, Berlin, Heidelberg.

[10] Li, F., Golden, B &, Wasil, E. (2007). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results, Computers and Operations Research, vol. 34, pp. 2918–2930.

[11] Repoussis, P. P., Tarantilis, C. D. & Ioannou, G. (2007). The open vehicle routing problem with time windows. Journal of the Operational Research Society., vol. 58, no. 3, pp. 355-367.

[12] Pisinger, D. & Ropke, S. (2007). A general heuristic for vehicle routing problems, Computers and Operations Research, vol. 34, pp. 2403– 2435.

[13] Tarantilis CD., Diakoulaki D. & Kiranoudis CT. (2004). Combination of geographical information system and efficient routing algorithms for real life distribution operations. European Journal of the Operational Research, vol. 152, pp. 437–53.

[14] Marinakis, Y. & Marinaki, M. (2014). Bumble Bees Mating Optimization algorithm for the Open Vehicle Routing Problem, Swarm and Evolutionary Computation, vol. 15, pp. 80–94.

[15] López-Sánchez, A. D., Hernández-Díaz, A. G. Vigo, D. Caballero, R. & Molina, J. (2014). A multi-start algorithm for a balanced real-world Open Vehicle Routing Problem, European Journal of Operational Research, vol. 238, no. 1, pp. 104-113.

[16] Brito, J., Martínez, F. J., Moreno, J. A. & Verdegay, J. L. (2015). An ACO hybrid metaheuristic for close–open vehicle routing problems with time windows and fuzzy constraints, Applied Soft Computing, vol. 32, pp. 154-163.

[17] Sevkli, A. Z. & Güler, B. (2017). A multi-phase oscillated variable neighbourhood search algorithm for a real-world open vehicle routing problem. Applied Soft Computing, vol. 58, pp. 128-144.

[18] Niu, Y., Yang, Z. & Chen, P., & Xiao, J. (2018). Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost. Journal of Cleaner Production, vol. 171, pp. 962-971.

[19] Dorrani, Z & Mahmoodi, M. S. (2016). Noisy images edge detection: Ant colony optimization algorithm. Journal of AI and Data Mining, vol. 4, no. 1, pp. 77-83.

[20] Larki, H. & Yousefikhoshbakht, M. (2014). Solving the multiple traveling salesman problem by a novel meta-heuristic algorithm. Journal of Optimization in Industrial Engineering, vol. 7, no. 16, pp. 55-63.

[21] Mahmoodi Darani, N., Ahmadi, V., Saadati Eskandari, Z. & Yousefikhoshbakht, M. (2013). Solving the Capacitated Clustering Problem by a Combined Meta-Heuristic Algorithm. Journal of Advances in Computer Research, vol. 4, no. 1, pp. 89-100.

[22] Yousefikhoshbakht, M., & Sedighpour, M. (2011). An optimization algorithm for the capacitated vehicle routing problem based on ant colony system. Australian Journal of Basic and Applied Sciences, vol. 5, no. 12, pp. 2729-2737.

[23] Peng, J., Li, H., Jiang, Q., Wang, Y. & Chen, J. (2014). An integrative approach for measuring semantic similarities using gene ontology. BMC systems biology, vol. 8, no. 5, S8.

[24] Christofides, N., Mingozzi, A. & Toth, P. (1979). The vehicle routing problem, in: N. Christofides, A. Mingozzi, P. Toth and C. Sandi (eds.), Combinatorial Optimization, Wiley, Chichester, pp. 315-338.

[25] Baker, B., & Ayechew, M. (2003). A genetic algorithm for the vehicle routing problem, Computers and Operations Research, vol. 30, pp. 787–800.

[26] Zhang, X. & Tang, L. (2009). A new hybrid ant colony optimization algorithm for the vehicle routing problem, Pattern Recognition Letters, vol. 30, no. 152, pp. 848–855.

[27] Ai, T. J. & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery, Computers & Operations Research, vol. 36, no. 5, pp. 1693- 1702.

[28] Marinakis, Y. & Marinaki, M. (2010). A hybrid genetic – particle swarm optimization algorithm for the vehicle routing problem, Expert Systems with Applications, vol. 37, no. 33, pp. 1146-1455.

[29] Fu, Z., Eglese, R. & Li, L. Y. O. (2005). A new tabu search heuristic for the open vehicle routing problem, Journal of the Operational Research Society, vol. 56, pp. 267–274.

[30] Li, F., Golden, B. & Wasil, E. (2007). The open vehicle routing problem: Algorithms, large-scale test problems, and computational results, Computers and Operations Research, vol. 34, pp. 2918–2930.

[31] Fleszar, K., Osman, I. H. & Hindi, K. S. (2009). A variable neighbourhood search algorithm for the open vehicle routing problem, European Journal of Operational Research, vol. 195, pp. 803–809.

نشریه هوش مصنوعی و داده کاوی

یک روش فراابتکاری ترکیبی برای حل مساله مسیریابی وسیله نقلیه و نسخه باز آن

مجید یوسفی خوشبخت [۱] و نرگس محمودی دارانی [۲،*]

[۱] گروه ریاضی، دانشکده علوم، دانشگاه بوعلی سینا، همدان، ایران.

[۲] گروه ریاضی، دانشگاه ازاد اسلامی، واحد هشتگرد، کرج، ایران.

ارسال ۲۰۱۸/۰۵/۲۷؛ بازنگری ۲۰۱۸/۰۸/۳۱؛ پذیرش ۲۰۱۸/۱۱/۱۴

چکیده:

مساله مسیریابی وسیله نقلیه باز (OVRP) یکی از مهمترین نسخه های مساله مسیریابی وسیله نقلیه (VRP) است که دارای بسیاری از کاربردها در صنعت و خدمات است. در VRP، مجموعه ای از مشتریان به همراه مقدار کالا مورد نیازشان داده شده است در حالی که یک ناوگان از وسایل نقلیه ظرفیت دار موجود است. به علاوه هزینه حرکت بین انبار و مشتریان نیز داده شده است. در مساله OVRP برخلاف VRP، وسایل نقلیه لازم نیست که به انبار کالا باز گردند. چون این دو مساله به مسایل NP-سخت تعلق دارند، در این مقاله یک الگوریتم ترکیبی نمونه مورچگان به نام EACO برای حل آنها ارایه شده است. در این الگوریتم برای بهبود بیشتر، الگوریتم جستجوی ممنوع، یک قانون جدید انتخاب و یک روش اصلاحی بروزرسانی فرمون ارایه گردیده است. براساس این اصلاحات، الگوریتم توانسته که از بهینه های محلی فرار کند و مناطق بیشتری از فضای نمونه را مورد جستجو قرار دهد. نتایج محاسباتی روی ۱۴ مثال برای این دو مساله نشان می دهد که الگوریتم EACO برای بیشتر مسایل به بهترین جواب ها دست پیدا کرده است و در مقایسه با دیگر الگوریتم های فراابتکاری در معیار کیفیت جواب ها بسیار مقایسه پذیر است.

کلمات کلیدی: مساله مسیریابی وسیله نقلیه، مساله مسیریابی وسیله نقلیه باز، الگوریتم نمونه مورچگان، جستجوی ممنوع، مسایل NP-سخت.