

# A Combination of Meta-heuristic and Heuristic Algorithms for the VRP, OVRP and VRP with Simultaneous Pickup and Delivery

*Maryam Ashouri*

Instructor, Department of Civil Engineering,  
Malayer Branch, Islamic Azad University, Malayer, Iran

*Majid Yousefikhoshbakht*

Assistant Professor, Department of Mathematics,  
Faculty of Science, Bu-Ali Sina University, Hamedan, Iran  
khoshbakht@basu.ac.ir

## Abstract

Vehicle routing problem (VRP) is a Nondeterministic Polynomial Hard combinatorial optimization problem to serve the consumers from central depots and returned back to the originated depots with given vehicles. Furthermore, two of the most important extensions of the VRPs are the open vehicle routing problem (OVRP) and VRP with simultaneous pickup and delivery (VRPSPD). In OVRP, the vehicles have not return to the depot after last visit and in VRPSPD, customers require simultaneous delivery and pick-up service. The aim of this paper is to present a combined effective ant colony optimization (CEACO) which includes sweep and several local search algorithms which is different with common ant colony optimization (ACO). An extensive numerical experiment is performed on benchmark problem instances addressed in the literature. The computational result shows that suggested CEACO approach not only presented a very satisfying scalability, but also was competitive with other meta-heuristic algorithms in the literature for solving VRP, OVRP and VRPSPD problems.

**Keywords:** Meta-heuristic algorithms, Vehicle Routing Problem, Open Vehicle Routing Problem, Simultaneously Pickup and Delivery, Ant Colony Optimization.

## 1. Introduction

The capacitated vehicle routing problem (CVRP) is one of the most important combinatorial optimization problems, which recently has been receiving much attention by researchers and scientists (Yousefikhoshbakht & Dolatnezhad, 2016). CVRP deals with servicing a set of delivery customers or a set of pickup customers by a set of vehicles stationed at a central depot. Each vehicle, visits a set of customers such that every customer is visited exactly once and by exactly one vehicle. Furthermore, the capacity of each vehicle must not be violated. The objective of CVRP is to plan a set of routes to service all customers while minimizing the total travel distance (Yousefikhoshbakht et al., 2014).

The open vehicle routing problem (OVRP) is one of the most important variant of VRP which has recently attracted the attention of scientists and researchers because of many applications in industrial and service firms specially delivering packages and newspapers to homes. This problem similar to VRP involves routing a homogeneous fleet of vehicles that start to move simultaneously from the depot but do not come back to the depot after visiting customers. Each vehicle has a fixed capacity and perhaps a route-length restriction which limits the maximum distance it can travel. Each customer has a known demand and is serviced by only one visit of a single vehicle. The objective is to design a set of minimum cost routes to serve all customers so that the load on a vehicle is below vehicle capacity at each point on the route. In addition, we need to find the minimum number of vehicles which are required to service all customers. The description of this important variant of the VRP appeared in the literature over 30 years ago in which contractors who are not employees of the delivery company use their own vehicles and do not return to the depot. As a result, researcher interest in the OVRP has increased dramatically and a

wide variety of new algorithms have been developed over the last ten years to solve the problem (Sedighpour et al., 2014). From the point of view of graph theory, the difference between the OVRP and the VRP is that a solution of the former consists of a set of Hamiltonian paths rather than Hamiltonian cycles. The problem of finding the best Hamiltonian path for each set of customers assigned to a vehicle is NP-hard (Yousefikhoshbakht et al., 2011). Hence OVRP is also NP-hard.

The vehicle routing problem with Simultaneous pickup and delivery (VRPPD) is another important variant of VRP. The major difference between this problem and VRP is that customers may receive or deliver goods simultaneously in VRPPD, while in VRP all customers just receive goods from a depot. Deliveries are supplied at the start of the vehicle's service, while pick-up loads are taken to the same depot at the end of the service. One important characteristic of this problem is that a vehicle's load in any given route is a mix of pick-up and delivery loads. The VRPPD belongs to the field of reverse logistics context from a practical point of view. This is gaining increasing importance due to more people becoming environmentally conscious. Besides, companies are increasingly faced with the task of managing the reverse flow of finished goods or raw-materials. They get involved with reverse logistics because they can economically profit from it; or they are obliged by legislation; or they feel socially impelled (de Brito and Dekker, 2004). In the context of this problem, the capacity of the vehicle cannot be exceeded. Furthermore, other constraints such as maximum distance or time windows may exist, may be considered in this problem. The objective is to find a set of routes to minimize the total travelling cost while meeting customer demands. One obvious difference among VRP and VRPPD is the variation of the vehicle load during the whole route. In VRP, the load decreases (increases) monotonously while in the VRPPD, the load of vehicles varies in a ruleless manner and the maximum may appear in the middle of the route. Furthermore, if the total demand of all the customers assigned to the same vehicle does not exceed the capacity limit in VRP, then the feasibility of the route would always behold, no matter what the visiting sequence is. However, it is quite different in the VRPPD.

The VRP holds a central place in logistics management and has been extensively studied in the operational research literature. Its great success can be attributed to the fact that it is a very interesting problem both from practical and theoretical points of view. Regarding practical point of view, the VRP definitely plays a central role in the efficiency of the operational planning level of distribution management, producing economical routes that contribute to reduction of distribution costs, offering simultaneously significant savings in all related expenses (capital, fuel costs, driver salaries, etc.). This problem was firstly defined by Dantzig and Ramser more than 50 years ago (Dantzig, and Ramser, 1959). Since then, it has had many applications in other problems. In contrast to the VRP, the OVRP has only been considered by a very limited number of researchers. From the early 1980s to the late 1990s, the OVRP received very little attention in the research operations in literature. However, several researchers have used some algorithms especially meta-heuristic ones since 2000. As far as we know, the first author to declare the OVRP was Schrage (1981) who raised for the first time the problem dedicated to the description of realistic routing problems, bringing attention to some of its applications. VRPPD was first proposed almost two decades ago from a distribution problem of a public library, with one depot, two vehicles and 22 customers (Min, 1989). This author presented a heuristic method to solve this real-life problem by using the below steps:

- (i) Customers are first clustered in such a way that the vehicle's capacity is not exceeded in each group.
- (ii) One vehicle is assigned to every cluster as Traveling Salesman Problem.
- (iii) The TSP is solved for each group of customers.

In the late sixties, few exact algorithms were developed for a very small number of variables and constraints because the exact algorithms could not perform well for large instances of VRP,

OVRP and VRPSPD. Therefore, nowadays many researchers have focused on meta-heuristic classes of algorithms. These algorithms strived for near optimal solutions in a reasonable computational cost, without however any guarantees of optimality. Currently, research activity in the field involves efforts that focus on finding modern meta-heuristics, producing high quality solutions for larger problems, which unfortunately are computationally extremely hard to be applied for exact algorithms (guaranteeing optimality).

For example, Zhao et al. proposed a hybrid quantum evolution algorithm for solving the VRP and showed that this algorithm is better than a genetic algorithm and a particle swarm optimization algorithm (Zhao et al., 2009). Teoh et al. combined neighborhood search with a differential evolution algorithm for solving the VRP and compared their algorithm with the cluster and searched heuristic algorithm and other meta-heuristic optimization algorithms to show its performance (Teoh et al., 2015). An effective hybrid quantum genetic algorithm to solve the VRP is proposed in Cai et al. (2010). In this algorithm, the immune operator is used to improve the local search ability of the algorithm. The author of (Bortfeldt, 2012) combined the powerful tabu search with a tree search algorithm for solving the VRP and compared their results to other meta-heuristics. The results show the superiority of the hybrid algorithm. Wang designed a cellular ant colony optimization algorithm in which the principle of cell automaton with the ant colony optimization search mechanism is combined to improve the algorithm's search capability (Wang, 2010).

Letchford et al. (2007) proposed a powerful meta-heuristic algorithm in order to solve OVRP in which some local search algorithms for examining wide solution neighborhoods are used. In their problem, two objective configurations including minimizing the number of routes and minimizing the routing cost of the generated route set are considered simultaneously. Furthermore, a hybrid meta-heuristic algorithm for solving the OVRP was proposed in (Repoussis et al., 2010). In this algorithm, the basic solution framework of evolutionary algorithms combined with a memory-based trajectory local search technique was utilized. The proposed method used arcs extracted from parent individuals and the selection and combination of arcs are dictated by a vector of strategy parameters. The results showed that the quality of each new offspring is further enhanced via a memory-based trajectory local search technique in the algorithm. Zachariadis & Kiranoudis proposed a local search algorithm whose moves are statically encoded into static move descriptor entities to explore the wide neighborhoods within the reasonable computational effort. In the local search operator applied to the candidate solution, only a limited solution part is modified. To diversify of the proposed algorithm, a tabu scheme for preventing iterated solution and a penalization strategy are employed. Mirhassani & Abolghasemi (2011) proposed a particular decoding technique for a real-value version of particle swarm optimization in order to apply to the OVRP. In this paper, a vector of the customer's position is constructed in descending order in the decoding method. Furthermore, the insert algorithm has been applied to constructed routes which seem promising to result in a better solution.

As same as VRP and OVRP, heuristics are thought to be more efficient and very popular for VRPSPD. For examples, Salhi and Nagy (1999) proposed four insertion heuristics and introduced the weak and strong feasibility of a solution for solving the VRPSPD. Emmanouil et al. proposed a hybrid algorithm based on tabu search and guided local search for this problem in 2009 (Emmanouil et al., 2009). The results show the efficiency of their meta-heuristic algorithm compared to other algorithm on benchmark instances involving from 50 to 400 customers. A large neighborhood search associated with a procedure similar is proposed to solve several variants of the VRP including VRPSPD (Ropke and Pisinger, 2004) and a modified ant colony algorithm is offered in which a new saving based visibility function and pheromone updating procedure (Catay, 2010).

Cruz et al. considered VRPSPD and applied a heuristic algorithm combined with cheapest insertion, cheapest insertion with multiple routes, variable neighborhood search, variable

neighborhood descent, tabu search, etc (Cruz et al., 2012). In this algorithm, first three procedures are used to obtain a high quality initial solution, and the last two algorithms are applied as local search methods. The results showed that not only the algorithm was competitive other famous algorithms, but also it was able to generate high quality solutions for some standard instances.

Goksal et al. presented a particle swarm optimization with a local search and a variable neighborhood descent algorithm (VND) for solving the VRPSPD (Goksal et al., 2013). The computational results indicate that the proposed algorithm competes with the heuristic approaches in the literature and improves several best known solutions. Finally, a class of extensions to VRP including different problem versions and some more well-known recent versions are proposed in Wassan and Nagi (2014). In this paper, a central focus was done on novel problem classes and an integer linear programming formulation was also presented. In addition, it is also shown how this formulation can be adapted to cater for other problem versions of VRP.

Finally, an applied version of VRP called heterogeneous VRPSPD was proposed in (Avci and Topaloglu, 2016). This problem, which can arise in many transportation systems involving both distribution and collection operations, is considered to be an NP-hard problem because it generalizes the classical VRP. Therefore, a hybrid local search algorithm in which a non-monotone threshold adjusting strategy is integrated with tabu search was proposed for this new problem. In order to test the efficiency of the proposed algorithm, a set of randomly generated problem instances was considered and the results showed the efficiency of the algorithm.

Because VRP, OVRP and VRPSPD are NP-hard combinatorial optimization problems, there are some serious difficulties in finding an exact solution in reasonable time. Therefore, many researchers have found that the employment of hybrid algorithms for optimization problems can improve the quality of solution results in comparison to heuristic and meta-heuristic approaches (Larki and Yousefikhoshbakht, 2014). In addition, ACO algorithm is one of the optimal algorithms for solving a routing scheme problem, and is also an effective method to find optimal solutions to difficult routing problems. Like many meta-heuristic algorithms, the primary ACO may fall into local minimum trap during the search process and it might get far from the global optimum. For this reason, we try to propose an effective two-phase ACO called CEACO by adopting a behavior between rigid regularity and randomness based on pure chance. At the first stage, these problems are solved by a modified ACO algorithm. At the second stage, the Lin-Kernighan algorithm is used for improving solutions, as this algorithm is one of the most effective local search algorithms. This algorithm leads to overcome premature local optima encountered and evolves towards diverse trajectories of the solution space, and a more extensive search is accomplished. In more details, to enhance the global exploration capability, the 2-opt local search with a high probability of  $\alpha$  and 0-1 and 1-1 exchanges with low probability  $\beta$  and  $\mu$  respectively are used in every iteration of ACO algorithm. The main contributions of the paper are as follows:

- (i) Presenting an effective ACO algorithm for three routing problems.
- (ii) Combining the ACO algorithm with a Lin-Kernighan algorithm.
- (iii) Presenting a new algorithm which is equipped with diversification and intensification mechanisms for solving the VRP, OVRP and VRPSPD.

The proposed framework is applied to 14 VRP, 16 OVRP and 28 VRPSPD benchmark instances derived from the literature and involving from 50 to 200 customers. The results show the proposed algorithm not only obtained several of the best known solutions, but also presented very satisfying solutions for other instances.

The rest of this paper is organized as follows. In section 2, the proposed CEACO are explained and its performance will be described. The computational results of the proposed algorithm are compared with some of the other algorithms on standard VRP, OVRP and VRPSPD problems in section 3. Finally, the conclusions are presented in section 4.

## 2. The proposed algorithm

Ant colony optimization (ACO) is one of the most popular meta-heuristic algorithms inspired by the behavior of real ants seeking a path between their colony and a source of food. For the first time, this algorithm was used to solve the traveling salesman problem (TSP) as shown in Figure 1.

- 1) Set the parameters.
- 2) Place  $m$  ants at the depot.
- 3) For each ant do below steps until there is not unvisited node:
  - Select a nest node based on formula (1).
  - Deposit pheromone on the obtained arcs.
- 4) Update the best solution and its obtained value until now.
- 5) If the stop condition is not satisfied, go to step 2.
- 6) Print the best solution and its value.

Figure 1. The ACO for the TSP

In the mentioned figure, the probability that any ant chooses one path over other increases as the amount of pheromone present increases. The decision for choosing the unvisited  $N_i$  node by ant  $k$  located at node  $i$  is made based on the formula (1) in which both  $a$  and  $b$  are parameters. In his formula,  $t_{ij}$  is amount of the pheromone of edge  $(i,j)$ ,  $h_{ij}$  is the inverse distance of edge  $(i,j)$  and ants release “pheromone information” ( $Dt_{ij}$ ) on the respective path while moving from node  $i$  to node  $j$  by formula  $t_{ij}(t) \leftarrow t_{ij}(t) + Dt_{ij}$ .

$$P_{ij}^k = \begin{cases} \frac{t_{ij}^a h_{ij}^b}{\sum_{j \in N_i^k} t_{ij}^a h_{ij}^b} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \quad (1)$$

This algorithm uses of pheromone evaporation in order to prevent the rapid convergence of ants to a sub-optimal path by formula  $t \leftarrow (1 - r)t$ . In the formula, pheromone density is reduced in each iteration by  $0 \leq r \leq 1$  which is set by the user. Besides,  $t$  is the matrix for the existing pheromone on the edges of the respective graph.

The ant system is the first version of ACO which could not produce acceptable results compared with meta-heuristic algorithms of the time. Therefore, several variants of ACO such as elite ant system (EAS), rank based ant system (RAS) and max-min ant system (MMAS) have been derived from the basic AS which are able to produce better results for many combinatorial optimization problems. In the proposed CEACO algorithm, if there are  $m$  vehicles in the problem, this number of ants is exactly positioned on  $n$  vertices randomly. Then ant  $k$  selects the next node  $j$  from the current node  $i$  among the unvisited nodes  $N_i^k$  by a new transition rule. This rule is used to better explore the feasible space and is obtained as follows:

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [\kappa_{ij}(t)]^\lambda}{\sum_{r \in J_i^k} [\tau_{ir}(t)]^\alpha [\eta_{ir}(t)]^\beta [\kappa_{ir}(t)]^\lambda} \quad \forall j \in J_i^k \quad (2)$$

where

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

$\eta_{ij}(t)$ : is the heuristic information for the ant visibility measure.

$\alpha, \beta, \lambda$  : Control parameters which changeable b user.

$k_{ij}$  : 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  $K_{ij} = c_{i0} + c_{0j} - c_{ij}$  where node 0 is the depot and  $c_{ij}$  denotes the distance between nodes  $i$  and  $j$ .

The CEACO mainly consists of the iteration of the three below steps

- 1) Each ant builds the solution independently for  $n$  groups
- 2) Apply the 2-opt, insert and move local search algorithms to improve the solutions
- 3) Update the global pheromone information.

The literature on meta-heuristics indicates that a promising approach for obtaining high-quality solutions is to couple a local search algorithm with a mechanism to generate initial solutions. So, after all ants have constructed their routes and before updating global pheromone, three types of local search schemes including 2- opt scheme, insert and swap moves are performed to further reduce the routes length (Figure 2). In insert algorithm, a customer is moved to another route but in swap algorithm a customer in a certain route is swapped with another customer from a different route. One of the most commonly encountered moves is the 2-opt which starts with a feasible tour and continues by omitting two arcs of the same route, which are not adjacent and then connects them again by another method in such a way that the new tour length is shorter. In multiple routes, two edges belong to different routes, which form a criss-cross, are selected and two new edges are replaced. It also should be noted that the new solution will be only accepted in state that first, the constraints are not violated specially about each vehicle's capacity. It can be noted that there are several routes for connecting nodes and producing the tour again, but a state that satisfies the problem's constraints is acceptable.

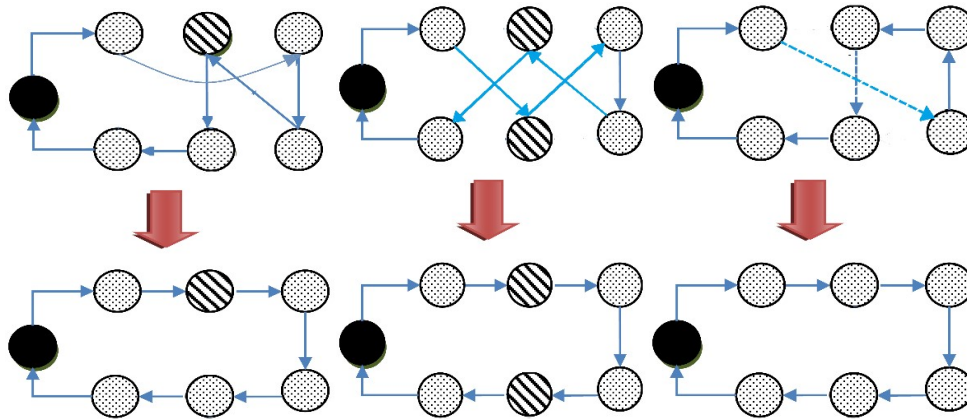


Figure 2. Insert (left), swap (middle) and 2-opt exchanges (right)

It is noted that the pheromone updating formula was used to simulate the change in the amount of pheromone due to both the addition of a new pheromone deposited by ants on the visited edges, and the pheromone evaporation. When all  $n$  groups of ants have completed their schedule, the pheromone level is updated by applying the global updating rule only on the paths that belong to the best found solutions since the beginning. This proposed algorithm ranks the solutions constructed by ants. What distinguishes this algorithm from the other algorithms is the fact that in CEACO the amount of releasing pheromone is based on the rank of the groups of ants in finding solutions by formula (3).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{\mu=1}^{\sigma} \Delta\tau_{ij}^{\mu}(t) \quad (3)$$

where:

$$\Delta\tau_{ij}^{\mu}(t) = \begin{cases} (\sigma - \mu + 1) \cdot \frac{Q}{L^{\mu}(t)} & (i, j) \in S^{\mu} \\ 0 & (i, j) \notin S^{\mu} \end{cases} \quad (4)$$

In this formula,  $Q$  is a constant coefficient determined by the user,  $\sigma$  is the number of groups of ants which have been ranked,  $\mu$  is the variable indicating ranking index from 1 to  $\sigma$  and  $S^{\mu}$  is the edges traversed by an ant group which has gained the  $\mu^{\text{th}}$  rank in finding the best solution. In our algorithm,  $\sigma$  best groups of ants of the algorithm have been allowed to lay pheromone on the arcs they traversed. The idea of the elitist strategy in the context of the proposed algorithm is to give extra emphasis to the best paths found so far in every iteration. This modification leads to balance between exploitation through emphasizing global best ant and exploration through the emphasis to iteration best ant.

Furthermore, when the algorithm attained a better solution compared to previous iterations, Lin-Kernighan algorithm is used to prevent the proposed algorithm from getting trapped in stagnation. In fact, the probability of finding better solutions near a good solution is relatively high. Since Lin-Kernighan algorithm is simple and has high quality to obtain sub-optimal or optimal solutions, this algorithm is used in this paper. Besides, in each step of the  $\omega$ -opt algorithm,  $\omega$  links of the current tour are replaced by  $\omega$  links in order to obtain a shorter tour. The time complexity of testing  $\omega$ -exchange is equal to  $O(n^{\omega})$  in which there is no nontrivial upper bound of the number of  $\omega$ -exchanges. The Lin-Kernighan algorithm introduced a powerful variable  $\omega$ -opt algorithm in which the value of  $\omega$  is changed during its execution. In this algorithm, a growing set of potential exchanges is considered which start with  $r = 2$  at each iteration. These exchanges are chosen in such a way that a feasible tour may be formed at any stage of the process. Then, the problem is examined for ascending values of  $\omega$  and the algorithm decides what the value of  $\omega$  should be. If an interchange of  $\omega$  links succeeds in finding a new shorter tour, then the actual tour is replaced with the new tour. This continues until some stopping conditions are satisfied. To save the computation time, we will only apply local search to the best current solution and best found solution until now in each iteration. The idea here is that a better solution may have a better chance to find a global optimum. After each iteration, if the best solution is not changed during 20 times, the algorithm finished and the best found solution is reported. Otherwise, the algorithm goes to transition rule step. Figure 3 shows the pseudo code of the proposed algorithm.

- 1) Set the parameters of the proposed algorithm, including  $\alpha, \beta, \lambda, \sigma, \mu$  and  $Q$ .
- 2) For  $n$  groups, place  $m$  ants at  $n$  vertices randomly.
- 3) Select a nest node for each ant based on a new proposed formula.
- 4) If there is a node that has not been visited, go to step 3.
- 5) Apply local search algorithms for obtained solutions in each iteration.
- 6) Save the best solution and its value obtained in current iteration.
- 7) Update global pheromone on each edge belonging to the ranked groups.
- 8) Apply Lin-Kernigan algorithm to the best solution in each iteration and best found solution until now.
- 8) Update the best solution and its value obtained until now.
- 10) If the best solution till now is improved within 15 iterations, go to step 2.
- 11) Show the best solution and its value.

Figure 3. The pseudo of CEACO

### 3. Numerical Analysis

In this section, the results of the proposed CEACO for solving the VRP instances are shown. At the second and third subsections, the proposed algorithm is used to solve OVRP and VRPSPD instances available in the literature. This algorithm is coded in Matlab 11 programming language and executed on a PC equipped with an Intel Pentium IV processor running at 3500 MHz; Core i3 and 8 GB of RAM running Microsoft Windows 7 Ultimate. Because the CEACO is a meta-heuristic algorithm, the best solution found for ten independent runs is reported.

#### 3.1. Application in VRP instances

In this subsection, the performance of the proposed algorithm is tested on a set of 14 benchmark instances with 50–199 customers which have been widely used as benchmarks. The first 10 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 test instances are subjected to capacity constraints while the problems 6–10, 13 and 14 have, also, the route length limitations. The information of the 14 instances is shown in Table 1. In this table, instance, n, m, and BKS are the names of instances, number of customers, number of vehicles and the best known solution by other algorithms (BKS). In this table, in order to assess the effectiveness of the CEACO, its results are compared to genetic algorithm (GA) (Baker and Ayechev, 2003), scatter search algorithm combined by ant colony optimization (SS\_ACO) (Zhang and Tang, 2009), particle swarm intelligent (PSO) (Ai, Kachitvichyanukul, 2009) and genetic algorithm and particle swarm intelligent (GAPSO) (Marinakis and Marinaki, 2010). As can be seen from this table, the proposed algorithm finds the optimal solution for 8 out of 14 problems and is a competitive algorithm compared to the BKS. Furthermore, the gap of other problems is about as high as 1% and so the proposed algorithm finds nearly the best known solutions.

Table 1. Computational results for standard VRP problems

| Instance | n   | m  | GA      | SS_ACO  | PSO     | GAPSO   | CEACO   | 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  | 837.29  | 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 | 1318.13 | 1291.45 |
| C6       | 50  | 6  | 560.29  | 559.12  | 555.43  | 555.43  | 555.43  | 555.43  |
| C7       | 75  | 11 | 914.13  | 912.68  | 917.68  | 909.68  | 909.68  | 909.68  |
| C8       | 100 | 9  | 872.82  | 869.34  | 867.01  | 865.94  | 865.94  | 865.94  |
| C9       | 150 | 14 | 1193.05 | 1179.4  | 1181.14 | 1163.41 | 1167.33 | 1162.55 |
| C10      | 199 | 18 | 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  | 830.83  | 819.56  |
| C13      | 120 | 11 | 1562.25 | 1556.52 | 1546.20 | 1544.57 | 1548.13 | 1541.14 |
| C14      | 100 | 11 | 872.34  | 870.26  | 866.37  | 866.37  | 866.37  | 866.37  |

Furthermore, the GA has not been able to find the best solutions in thirteen of the fourteen examples. Therefore, it is the weakest algorithm among the five presented algorithms. However, SS\_ACO has been able to find better solutions than the GA and has come up with the best solution in 12 examples. Among remaining five algorithms, PSO has failed in improving the solutions in 10 examples and has come up with solutions similar to the ones found by GA. From the comparison

between GAPS0 and CEACO, it can be seen that GAPS0 in six examples has been able to find better solutions than the proposed algorithm. However, the CEACO has found better solutions than this algorithm for one example. For example, the solution of C6 is shown in Figure 4 which is the best found solution until now by other algorithms.

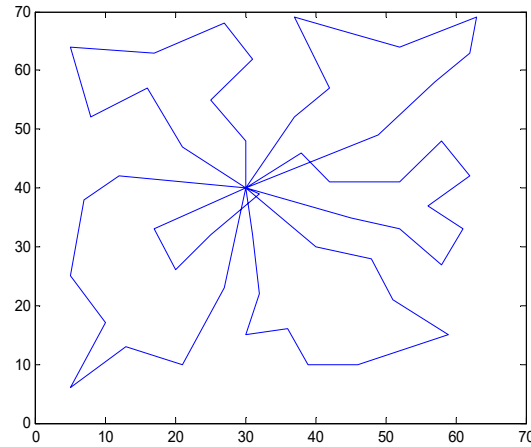


Figure 4. The solution of C6 found by CEACO

### 3.2. Application in OVRP instances

In this subsection, 16 instances of OVRP which the number of customers ranges in size from 50 to 199 are considered. In all instances, the vertices are in the Euclidean plane and the cost of an edge is equal to the Euclidean distance between its end-vertices computed with real numbers. These problems are identified by their original number and prefixed respectively with the letters C and F available in the literature and they are summarized in Table 2. The problems C1–C5, C11, C12, F11 and F12 have no driving time constraint, and C6–C10, C13 and C14 are the same instances as C1–C5, C11 and C12, but with a travel time constraint. In this table, some of the characteristics of these problems are described. The first column includes the instance name and the second and third columns show the number of customers  $n$ , and the number of used vehicles  $K$ . It is noted that this number of vehicles is fixed at the minimum possible for all of these instances estimated through dividing the sum of all customer demands by vehicle capacity. Furthermore, the fourth column shows the maximum route length which each vehicle can pass. The objective of the computational experiments is to compare the performance of the proposed algorithm with six below meta-heuristic algorithms for solving OVRP:

- Tabu search by Fu *et al.* (2005; 2006) (TSF)
- Tabu search by Brandão *et al.* (2004) (TSB)
- Tabu search by Brandão *et al.* (2004) (TSAN)
- Record-to-record travel algorithm by Li *et al.* (2007) (ORTA)
- Variable neighborhood Search by Fleszar *et al.* (2009) (VNS)
- Adaptive large neighborhood search (ALNS) by Pisinger & Ropke (2007)

This table indicates that some algorithms used different number of vehicles shown in the brackets. Furthermore, the last column shows best known solution (BKS) by the various algorithms until now. In the proposed algorithm, the minimum number of vehicles as specified by the lower bound of  $K$  is used. It is noted that some authors consider the distances as floating point numbers in their algorithms and then report the cost of the solutions with a fixed precision like one decimal place. In this paper, results only for the floating point versions are reported like other papers. Table 2 shows that CEACO finds the optimal solution for 10 out of 16 problems published in the literature. Moreover, TSB, TSF, TSAN, ORTR, ALNS and VNS have been able to find 0, 0, 0, 5, 4 and 6 optimal solutions from among these instances. These results indicate that the proposed

algorithm not only can obtain high quality solutions for all instances but also it is a competitive approach compared to these six algorithms. It should be noted that direct comparisons of the required computational times cannot be conducted as they closely depend on various factors such as the programming languages, the coding abilities of the programmers, the processing power of the computers, the compilers and the running processes on the computers.

Table 2. Results of the proposed algorithm compared to other algorithms

| Instance | n   | K  | L   | TSF        | TSB    | TSAN       | ORTR        | VNS    | ALNS        | CEACO       | BKS         |
|----------|-----|----|-----|------------|--------|------------|-------------|--------|-------------|-------------|-------------|
| C1       | 50  | 5  | -   | 416.1      | 416.1  | 438.2      | 416.06      | 416.06 | 408.5       | 408.5       | 408.5       |
| C2       | 75  | 10 | -   | 567.1      | 574.5  | 584.7      | 567.14      | 567.14 | 564.06 [11] | 564.06      | 564.06 [11] |
| C3       | 100 | 8  | -   | 643.1      | 641.6  | 643.4      | 639.74      | 639.74 | 617         | 617         | 617         |
| C4       | 150 | 12 | -   | 738.9      | 740.8  | 767.4      | 733.13      | 733.13 | 733.13      | 738.22      | 733.13      |
| C5       | 199 | 16 | -   | 879.0 [17] | 953.4  | 1010.9     | 924.96      | 905.96 | 870.26 [17] | 875.11      | 870.26 [17] |
| C6       | 50  | 5  | 180 | 413.0      | 412.96 | 416.0      | 412.96      | 412.96 | 400.6 [6]   | 400.6       | 400.6 [6]   |
| C7       | 75  | 10 | 144 | 568.5 [11] | 634.54 | 581.0 [11] | 568.49 [11] | 596.47 | 560.4 [11]  | 560.4 [11]  | 560.4 [11]  |
| C8       | 100 | 8  | 207 | 647.3      | 644.63 | 652.1      | 644.63      | 644.63 | 638.2 [9]   | 647.67      | 638.2 [9]   |
| C9       | 150 | 12 | 180 | 761.3 [14] | 785.2  | 827.6 [14] | 756.38 [14] | 760.06 | 752.0 [14]  | 757.92      | 752.0 [14]  |
| C10      | 199 | 16 | 180 | 903.1      | 884.63 | 946.8      | 876.02      | 875.67 | 875.67 [17] | 903.10      | 875.67 [17] |
| C11      | 120 | 7  | -   | 724.5      | 683.4  | 713.3      | 682.54      | 682.12 | 679.60 [9]  | 679.60      | 679.60 [9]  |
| C12      | 100 | 10 | -   | 534.7      | 535.1  | 543.2      | 534.24      | 534.24 | 534.24      | 534.24      | 534.24      |
| C13      | 120 | 7  | 648 | 922.3 [12] | 943.66 | 994.3      | 896.50 [12] | 904.04 | 896.50 [12] | 896.50 [12] | 896.50 [12] |
| C14      | 100 | 10 | 936 | 600.7      | 597.3  | 651.9 [12] | 591.87      | 591.87 | 591.87 [11] | 599.83      | 591.87 [11] |
| F11      | 71  | 4  | -   | 177.0      | 177.0  | 179.5      | 177.00      | 178.09 | 175.0       | 175.00      | 175.0       |
| F12      | 134 | 7  | -   | 778.6      | 781.2  | 825.9      | 769.66      | 769.66 | 769.66      | 769.66      | 769.66      |

### 3.3. Application in VRPSD instances

In this sub-section, the computational experiment is conducted on the Dethloff benchmark data set classified into SCA3, SCA8, CON3 and CON8 sets. In these data sets, SCA instances are generated with customers scattered uniformly in the service region of 100\_100 and CON instances are produced with half of the customers located uniformly in the service region and the other half are concentrated in the interval [100/3,100/3]. The numbers 3 and 8 after SCA or CON indicate the parameter for determining vehicle capacity. The characteristic of these algorithms are shown in table 3 as details. In this table,  $C$  is the number of customers,  $V$  represents the number of vehicles and BKS indicates the best solution found on the Web. Besides, the efficiency and performance of the proposed algorithm are compared with tabu Search proposed by Tang and Galvao in 2006 (TS) and large neighborhood search proposed by Ropke and Pisinger in 2006 (LNS). It should be noted that reported results for each instance is the best one over multiple runs, while LNS, TS and the proposed algorithm run 10 times for each instance. The reason considering TS and LNS in the comparison to CEACO is to draw a conclusion about performances of these meta-heuristics in VRPSD. It is also important to test the performance of the proposed algorithm against the TS and

LNS since these meta-heuristic algorithms are the most effective algorithms recently proposed in the literature. Finally, the results of CEACO shows the best found solution by the proposed algorithm in ten time iteration and Gap is the deviation of the proposed CEACO with respect to the BKS. The results show that among the 40 instances, the CEACO has produced the high quality results of 5 instances and equaled another 35. Therefore, the proposed algorithm demonstrated a consistent performance, since the average gap between the best solutions obtained by CEACO and the BKS solutions was only 0.05% with the highest value in all the instances.

Table 3. Results obtained in Dethloff's instances

| Problem | C  | V         | CEACO          | TS            | LNS           | BKS            | Gap  |
|---------|----|-----------|----------------|---------------|---------------|----------------|------|
| SCA3-0  | 50 | 4         | <b>635.62</b>  | 640.55        | 636.1         | <b>635.62</b>  | 0    |
| SCA3-1  | 50 | 4         | <b>697.84</b>  | <b>697.84</b> | <b>697.8</b>  | <b>697.84</b>  | 0    |
| SCA3-2  | 50 | 4         | <b>659.34</b>  | <b>659.34</b> | <b>659.3</b>  | <b>659.34</b>  | 0    |
| SCA3-3  | 50 | 4         | <b>680.04</b>  | <b>680.04</b> | 680.6         | <b>680.04</b>  | 0    |
| SCA3-4  | 50 | 4         | <b>690.50</b>  | <b>690.50</b> | <b>690.5</b>  | <b>690.50</b>  | 0    |
| SCA3-5  | 50 | 4         | <b>659.90</b>  | <b>659.90</b> | <b>659.9</b>  | <b>659.90</b>  | 0    |
| SCA3-6  | 50 | 4         | <b>651.09</b>  | 653.81        | <b>651.1</b>  | <b>651.09</b>  | 0    |
| SCA3-7  | 50 | 4         | <b>659.17</b>  | <b>659.17</b> | 666.1         | <b>659.17</b>  | 0    |
| SCA3-8  | 50 | 4         | <b>719.47</b>  | <b>719.47</b> | <b>719.5</b>  | <b>719.47</b>  | 0    |
| SCA3-9  | 50 | 4         | <b>681.00</b>  | <b>681</b>    | <b>681.0</b>  | <b>681.00</b>  | 0    |
| SCA8-0  | 50 | <b>9</b>  | <b>961.50</b>  | 981.47        | 975.1         | <b>961.50</b>  | 0    |
| SCA8-1  | 50 | <b>9</b>  | <b>1049.65</b> | 1077.44       | 1052.4        | <b>1049.65</b> | 0    |
| SCA8-2  | 50 | <b>9</b>  | 1042.75        | 1050.98       | 1044.5        | <b>1039.64</b> | 0.29 |
| SCA8-3  | 50 | <b>9</b>  | <b>983.34</b>  | <b>983.34</b> | 999.1         | <b>983.34</b>  | 0    |
| SCA8-4  | 50 | <b>9</b>  | <b>1065.49</b> | 1073.46       | <b>1065.5</b> | <b>1065.49</b> | 0    |
| SCA8-5  | 50 | <b>9</b>  | <b>1027.08</b> | 1047.24       | <b>1027.1</b> | <b>1027.08</b> | 0    |
| SCA8-6  | 50 | <b>9</b>  | <b>971.82</b>  | 995.59        | 977.0         | <b>971.82</b>  | 0    |
| SCA8-7  | 50 | <b>10</b> | <b>1051.28</b> | 1068.56       | 1061.0        | <b>1051.28</b> | 0    |
| SCA8-8  | 50 | <b>9</b>  | <b>1071.18</b> | 1080.58       | <b>1071.2</b> | <b>1071.18</b> | 0    |
| SCA8-9  | 50 | <b>9</b>  | 1063.14        | 1084.8        | <b>1060.5</b> | <b>1060.5</b>  | 0.25 |
| CON3-0  | 50 | 4         | <b>616.52</b>  | 631.39        | <b>616.5</b>  | <b>616.52</b>  | 0    |
| CON3-1  | 50 | 4         | <b>554.47</b>  | <b>554.47</b> | <b>554.5</b>  | <b>554.47</b>  | 0    |
| CON3-2  | 50 | 4         | 519.99         | 522.86        | 521.4         | <b>518.00</b>  | 0.36 |
| CON3-3  | 50 | 4         | <b>591.19</b>  | <b>591.19</b> | <b>591.2</b>  | <b>591.19</b>  | 0    |
| CON3-4  | 50 | 4         | <b>588.79</b>  | 591.12        | <b>588.8</b>  | <b>588.79</b>  | 0    |
| CON3-5  | 50 | 4         | <b>563.70</b>  | <b>563.70</b> | <b>563.7</b>  | <b>563.70</b>  | 0    |
| CON3-6  | 50 | 4         | <b>499.05</b>  | 506.19        | 500.8         | <b>499.05</b>  | 0    |
| CON3-7  | 50 | 4         | <b>576.48</b>  | 577.68        | <b>576.5</b>  | <b>576.48</b>  | 0    |
| CON3-8  | 50 | 4         | <b>523.05</b>  | <b>523.05</b> | <b>523.1</b>  | <b>523.05</b>  | 0    |
| CON3-9  | 50 | 4         | <b>578.25</b>  | 580.05        | 586.4         | <b>578.25</b>  | 0    |
| CON8-0  | 50 | <b>9</b>  | 860.48         | 860.48        | <b>857.2</b>  | <b>857.17</b>  | 0.39 |
| CON8-1  | 50 | <b>9</b>  | <b>740.85</b>  | <b>740.85</b> | <b>740.9</b>  | <b>740.85</b>  | 0    |
| CON8-2  | 50 | <b>9</b>  | <b>712.89</b>  | 723.32        | 716.0         | <b>712.89</b>  | 0    |
| CON8-3  | 50 | <b>9</b>  | <b>811.07</b>  | 811.23        | <b>811.1</b>  | <b>811.07</b>  | 0    |
| CON8-4  | 50 | <b>9</b>  | <b>772.25</b>  | <b>772.25</b> | <b>772.3</b>  | <b>772.25</b>  | 0    |
| CON8-5  | 50 | <b>9</b>  | <b>754.88</b>  | 756.91        | 755.7         | <b>754.88</b>  | 0    |
| CON8-6  | 50 | <b>9</b>  | <b>678.92</b>  | <b>678.92</b> | 693.1         | <b>678.92</b>  | 0    |
| CON8-7  | 50 | <b>9</b>  | <b>811.96</b>  | 814.5         | 814.8         | <b>811.96</b>  | 0    |
| CON8-8  | 50 | <b>9</b>  | 771.62         | 775.59        | 774.0         | <b>767.53</b>  | 0.53 |
| CON8-9  | 50 | <b>9</b>  | <b>809.00</b>  | <b>809</b>    | 809.3         | <b>809.00</b>  | 0    |

As can be seen in this table, the proposed algorithm is a competitive approach compared to the LNS and TS. On the other hand, it is seen that LNS and TS fail to reach the best results for 18 and 23 out of 40 instances, respectively. In 18 instances, both the CEACO and TS are same but, the

proposed algorithm finds a better solution than TS in other instances. We also conclude that the proposed method has better solution than the LNS for eighteen instances and equal solution for twenty one instances. Therefore, the best algorithm is EACO among these three algorithms which has found the best known solutions in 87%.

In order to test performance of the CEACO, the second data set of problem instances consist of 50 to 199 customers is suggested by Salhi and Nagy (1999). The cost matrix is found by calculating the Euclidean distances between vertices. Characteristic of these instances is shown in Table 3 in details. In this table, Columns 1-4 show the name of instance, problem size C, number of vehicles V, and the best value result of the CEACO. Besides columns 5-9 present results of ant colony system (ACS) (Gajpal & Abad, 2009), particle swarm optimization (PSO) (Ai & Kachitvichyanukul, 2009a), parallel iterative local search (PILS) (Subramanian et al., 2010), a mixed Insertion-based heuristics (MA) (Dethloff, 2001) and reactive tabu search (RTS) (Wassan et al., 2008).

Table 4. Comparison between EACO and other meta-heuristic algorithms

| Instance | C   | V  | EACO           | ACS     | PSO  | PILS    | MA   | RTS     | BKS     |
|----------|-----|----|----------------|---------|------|---------|------|---------|---------|
| CMT1X    | 50  | 3  | <b>466.77</b>  | 466.77  | 467  | 466.77  | 501  | 468.30  | 466.77  |
| CMT2X    | 75  | 6  | <b>668.77</b>  | 684.21  | 710  | 684.21  | 782  | 668.77  | 668.77  |
| CMT3X    | 100 | 5  | <b>721.27</b>  | 721.40  | 738  | 721.27  | 847  | 729.63  | 721.27  |
| CMT4X    | 150 | 7  | <b>852.46</b>  | 854.12  | 912  | 852.46  | 1050 | 876.50  | 852.46  |
| CMT5X    | 199 | 10 | <b>1029.25</b> | 1034.87 | 1167 | 1029.25 | 1348 | 1044.51 | 1029.25 |
| CMT6X    | 50  | 6  | <b>556.06</b>  | -       | -    | -       | 584  | 556.06  | 556.06  |
| CMT7X    | 75  | 11 | <b>901.22</b>  | -       | -    | -       | 961  | 903.05  | 901.22  |
| CMT8X    | 100 | 9  | <b>865.51</b>  | -       | -    | -       | 928  | 879.60  | 865.51  |
| CMT9X    | 150 | 15 | 1199.11        | -       | -    | -       | 1299 | 1220.00 | 1173.44 |
| CMT10X   | 199 | 19 | <b>1424.06</b> | -       | -    | -       | 1571 | 1464.58 | 1424.06 |
| CMT11X   | 120 | 4  | <b>835.26</b>  | 839.66  | 895  | 833.92  | 959  | 861.97  | 835.26  |
| CMT12X   | 100 | 5  | <b>644.70</b>  | 663.01  | 691  | 662.22  | 804  | 844.70  | 644.70  |
| CMT13X   | 120 | 11 | 1553.89        | -       | -    | -       | 1576 | 1647.51 | 1551.25 |
| CMT14X   | 100 | 10 | 822.97         | -       | -    | -       | 871  | 823.95  | 821.75  |
| CMT1Y    | 50  | 3  | <b>466.77</b>  | 466.77  | 467  | 466.77  | 501  | 458.96  | 466.77  |
| CMT2Y    | 75  | 6  | <b>663.25</b>  | 684.94  | 710  | 684.21  | 782  | 663.25  | 663.25  |
| CMT3Y    | 100 | 5  | <b>721.27</b>  | 721.40  | 740  | 721.27  | 847  | 745.46  | 721.27  |
| CMT4Y    | 150 | 7  | <b>852.35</b>  | 855.76  | 913  | 852.46  | 1050 | 870.44  | 852.35  |
| CMT5Y    | 199 | 10 | 1037.34        | 1037.34 | 1142 | 1029.25 | 1348 | 1054.46 | 1029.25 |
| CMT6Y    | 50  | 6  | <b>556.68</b>  | -       | -    | -       | 584  | 558.17  | 556.68  |
| CMT7Y    | 75  | 11 | <b>901.22</b>  | -       | -    | -       | 961  | 903.36  | 901.22  |
| CMT8Y    | 100 | 9  | <b>865.68</b>  | -       | -    | -       | 936  | 917.42  | 865.68  |
| CMT9Y    | 150 | 15 | 1192.74        | -       | -    | -       | 1299 | 1213.11 | 1171.95 |
| CMT10Y   | 199 | 19 | 1439.28        | -       | -    | -       | 1571 | 1419.79 | 1419.79 |
| CMT11Y   | 120 | 4  | <b>830.39</b>  | 840.19  | 900  | 833.92  | 1070 | 830.39  | 830.39  |
| CMT12Y   | 100 | 5a | <b>659.52</b>  | 663.50  | 697  | 662.22  | 825  | 659.52  | 659.52  |
| CMT13Y   | 120 | 11 | 1575.67        | -       | -    | -       | 1576 | 1647.04 | 1547.75 |
| CMT14Y   | 100 | 10 | <b>822.35</b>  | -       | -    | -       | 871  | 823.34  | 822.35  |

Finally, the column 10 shows the best known solutions (BKS). This table shows that the proposed algorithm can be used to solve VRPSPD effectively and the CEACO has produced the high quality results of 21 instances among the 28 instances. Furthermore, it can be seen that among these test problems, the maximum relative error is around 2% and the average relative error is less than 0.5%. In more details, the results show that the MA and PSO are not powerful algorithms for solving instances of VRPSPD and gains worse or equal solutions compared to the CEACO for all

instances. Besides, the CEACO gains worse solutions than the RTS in CMT1Y and CMT10Y, but it gains better or equal solutions than this algorithm in other problems. Furthermore, the results indicate that although the PILS obtains a better solution than the CEACO for CMT11X, this algorithm cannot maintain this advantage in the rest of the examples and the proposed algorithm yields better or equal solutions than this algorithm for other instances. Moreover, the computational experiments also show that in general the proposed algorithm produces better results compared to ACS algorithms in terms of the quality of the solution and could find better solutions for 9 of the 14 instances.

#### 4. Conclusion

In this paper, we have proposed a combined effective ant colony optimization (CEACO) for solving VRP, OVRP and VRPSDP problems. In this algorithm, a new state transition rule, a pheromone updating rule and three local search methods are used to improve the ACO in order to find better solutions. Furthermore, the Lin-Kernigan local search algorithm is applied when the best found solution until now is changed during running algorithm. A wide numerical experiment is performed on benchmark of these problems available in the literature. Computational results demonstrate that our algorithm has proven to be highly competitive in terms of quality of the solutions and is effective compared to other meta-heuristics. It seems that combining the proposed algorithm with strong other meta-heuristic algorithms can lead to better results for the proposed algorithm. Besides, some complex VRPSDP with more constraints such as time windows and multi-depots are a lot of research work in this field.

#### Acknowledgement

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

#### References

- Ai, T. J. & Kachitvichyanukul, V. (2009). *A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery*, Computers & Operations Research, 36(5), 1693- 1702.
- Avci, M. & Topaloglu, S. (2016). *A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery*, Expert Systems with Applications, 53, 160-171.
- Baker, B., and Ayechev, M. (2003). *A genetic algorithm for the vehicle routing problem*. Computers and Operations Research, Vol. 30, pp. 787–800.
- Bortfeldt, A. (2012). *A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints*. Computers & Operations Research, 39(9), pp. 2248-2257.
- Cai, B.B. & Zhang, X. H. (2010). Hybrid Quantum Genetic Algorithm and Its Application in VRP, journal of Computer Simulation, 7, p.068.
- Catay, B. (2010). *A new saving based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery*, Expert Systems with Applications, 37, 6809–6817.
- Cruz, R. C., Silva, T. C. B. D., Souza, M. J., Coelho, V. N., Mine, M. T., & Martins, A. X. (2012). GENVNS-TS-CL-PR: *A heuristic approach for solving the vehicle routing problem with simultaneous pickup and delivery*. Electronic Notes in Discrete Mathematics, 39, 217-224.
- Dethloff, J. (2001). *Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up*, OR Spektrum, 23(1), 79-96.
- Emmanouil, E. Z., Christos, D. T., & Chris. T. K. (2009). *A hybrid meta-heuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service*, Expert Systems with Applications, 36(2), 1070-1081.

- Goksal, F. P., Karaoglan, I. & Altiparmak, F. (2013). *A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery*, Computers & Industrial Engineering, 65(1), 39-53.
- Larki, H. & Yousefikhoshbakht, M. (2014). *Solving the multiple traveling salesman problem by a novel meta-heuristic algorithm*, Journal of Optimization in Industrial Engineering, 7(16), 55-63.
- Letchford, A.N., Lysgaard, J., & Eglese, R.W. (2007). *A Branch-and-cut Algorithm for the Capacitated Open Vehicle Routing Problem*. Journal of the Operational Research Society 58: 1642-1651.
- 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.
- MirHassani, S.A. & Abolghasemi, N. (2011). *A Particle Swarm Optimization Algorithm for Open Vehicle Routing Problem*. Expert Systems with Applications 38(9): 11547-11551.
- Repoussis, P.P., Tarantilis, C.D., Bräysy, O., & Ioannou, G. (2010). *A Hybrid Evolution Strategy for the Open Vehicle Routing Problem*. Computers & Operations Research 37 (3): 443-455.
- Ropke, S. & Pisinger, D. (2004). *A unified heuristic for a large class of vehicle routing problems with backhauls*, Technical Report 2004/14, University of Copenhagen.
- Salhi, S. & Nagy, G. (1999). *A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling*. Journal of the Operational Research Society, 50(10), 1034-1042.
- Schrage, L. (1981) *Formulation and structure of more complex/ realistic routing and scheduling problems*, Networks, 11, 229–232.
- Sedighpour, M., Ahmadi, V., Yousefikhoshbakht, M., Didehvar, F., & Rahmati, F. (2014). *Solving the open vehicle routing problem by a hybrid ant colony optimization*. Kuwait Journal of Science, 41(3), 139-162.
- Subramanian, A., Drummonda, L. M. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). *A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery*. Computers & Operations Research, 37, 1899–1911.
- Tang, F. A. & Galvao, R. D. (2006) *A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service*. Computers and Operations Research, 33(3), 595–619.
- Teoh, B. E., Ponnambalam, S. G., & Kanagaraj, G. (2015). *Differential evolution algorithm with local search for capacitated vehicle routing problem*. International Journal of Bio-Inspired Computation, 7(5), pp.321-342.
- Wang, Y. (2010), *Solving capacitated vehicle routing problem by cellular ant algorithm*, Journal of Information & Computational Science, 9, 2295-2304.
- Wassan, N. A., Wassan, A. H., & Nagy, G. (2008). *A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries*. Journal of Combinatorial Optimization, 15(4), 368–386.
- Yousefikhoshbakht, M. & Dolatnejad, A. (2016) *An Efficient Combined Meta-Heuristic Algorithm for Solving the Traveling Salesman Problem*. BRAIN, Broad Research in Artificial Intelligence and Neuroscience, 7(3), pp.125-138.
- Yousefikhoshbakht, M., Didehvar, F. & Rahmati, F. (2014). *An efficient solution for the vrp by using a hybrid elite ant system*. International Journal of Computers Communications & Control, 9(3), pp.340-347.

- Yousefikhoshbakht, M., Sedighpour, M., & Mahmoodabadi, E. (2011). *A Modified Elite ACO Based Avoiding Premature Convergence for Traveling Salesman Problem*. Journal of Industrial Engineering International, 7(15), pp. 68-75.
- Zachariadis, E. E. & Kiranoudis, C. T. (2010). *An Open Vehicle Routing Problem Metaheuristic for Examining Wide Solution Neighborhoods*. Computers & Operations Research 37(4): 712-723.
- 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.
- Zhao, Y.W., Peng, D.J., Zhang, J.L., & Wu, B. (2009). *Quantum evolutionary algorithm for capacitated vehicle routing problem*. Systems Engineering-Theory & Practice, 29(2), pp.159-166.



**Majid Yousefikhoshbakht** was born in Iran, in 1982. He received his B.Sc. degree in Mathematics from Bu-Ali Sina University, Hamedan, Iran, in 2006, M.Sc. and Ph.D. degrees from Amirkabir University of Technology, Tehran, Iran, in 2007 and 2014, respectively. Currently, he is an Assistant Professor in the Department of the Mathematics, Bu-Ali Sina University, Hamedan, Iran. His research interests include combinatorial optimization problems and artificial intelligence.