Kuwait J. Sci. **41** (3) pp. 139-162, 2014

Solving the open vehicle routing problem by a hybrid ant colony optimization

MOHAMMAD SEDIGHPOUR*, VAHID. AHMADI**,
MAJID YOUSEFIKHOSHBAKHT***, FARZAD DIDEHVAR***
AND FARHAD RAHMATI***

** Hamedan Branch, Islamic Azad University, Hamedan, Iran, sedighpour@iauh.ac.ir,*
*** Ahvaz Branch, Islamic Azad University, Ahvaz, Iran*
**** Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*

ABSTRACT

The open vehicle routing problem (OVRP) is a variant of vehicle routing problem (VRP) in which the vehicles are not required to return to the depot after completing a service. Since this problem belongs to NP-hard Problems, many metaheuristic approaches like ant colony optimization (ACO) have been used to solve it in recent years. The ACO has some shortcomings like its slow computing speed and local-convergence. Therefore, in this paper a hybrid ant colony optimization called HACO is proposed in which a new state transition rule, an efficient candidate list, several effective local search techniques and a new pheromone updating rule are used in order to achieve better solutions. Experimentation shows that the algorithm is successful in finding solutions within almost 3% of known optimal solutions on classical thirty one benchmark instances. Additionally, it shows that the proposed algorithm HACO finds twenty one best solutions of classical instances and is competitive with eight existing algorithms for solving OVRP. Furthermore, the size of the candidate lists used within the algorithm is a major factor in finding improved solutions and the computational times for the algorithm compare satisfactorily with other solution methods.

Keywords: Ant colony optimization; candidate list; local search techniques; np-hard Problems; open vehicle routing problem.

INTRODUCTION

The open vehicle routing problem (OVRP) is an important variant of vehicle routing problem (VRP) which has many applications in industrial and service firms. This problem has a unique character in that the vehicles are not required to return to the depot after completing a service. Although the OVRP has just recently attracted the attention of scientists and researchers, the description of this important variant of the VRP appeared in the literature over 30 years ago. The OVRP is utilized in practice in delivering packages and newspapers to homes. Therefore, it is one of the many extensions of the VRP used in industrial and service applications. In this problem, contractors who are not employees of the delivery company use their own vehicles and do not return to the depot. Furthermore, companies which use contractors to deliver newspapers to residential customers do not require the contractors and their vehicles to 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.

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 (Saadati & Yousefikhoshbakht, 2012). In other words, each route in the OVRP is a Hamiltonian path over the subset of customers visited on the route. 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. 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 (Syslo *et al.*, 1983). Hence OVRP is also NP-hard.

At first sight, having open routes instead of closed ones looks like a minor modification. Indeed, if travel costs are asymmetric, there is essentially no difference between the open and closed versions. To transform the open version into the closed one, it suffices to set the cost to zero for traveling from any customer to the depot. Indeed, somewhat surprisingly, it has been proved that the open version turns out to be more general than the closed one, in the sense that any closed VRP on n customers can be transformed into an open VRP on n customers, but there is no transformation in the reverse direction.

As it was mentioned before, the OVRP is a major problem faced by many distribution systems. In OVRP, like third-party logistics, when companies lease vehicles, the school bus and so on, the vehicles do not need to return to the depot in many cases. Therefore, it has attracted significant research attention and a number of algorithms have been proposed for its solution. Since there is no known polynomial algorithm that will find the optimal solution in every instance, the OVRP is considered NP-hard. For such problems, the use of heuristics such as ant colony optimization (ACO) is considered a reasonable approach in finding solutions. Moreover, although ACO has been successfully applied to several combinatorial optimization problems, it has some shortcomings like its slow computing speed and local-convergence (Yousefikhoshbakht & Sedighpour, 2012). Furthermore, because the OVRP is really difficult, the basic ACO algorithms cannot be directly applied to the problem with acceptable performance and few researchers have proposed new methods to improve the original ACO. Besides, although the development of modern metaheuristics has led to considerable progress, each metaheuristic has its own strengths and weaknesses. Therefore, many research studies have tried to develop hybrid algorithms, expecting to achieve the effectiveness and efficiency. In this paper, we have proposed an efficient hybrid ant colony optimization called HACO in order to improve both the performance of the algorithm and the quality of the solutions. The proposed algorithm took advantage of various versions of ant colony optimization algorithms for solving OVRP and then improved the global ability of the algorithm through importing new probability function of movement for constructing solutions, using new candidate list and updating pheromone method and several effective local searches. Therefore, the hybrid algorithm explores different parts of the solution space, and the search method is not trapped at the local optimum. The experimental results have shown that the HACO algorithm is to be very efficient and competitive in terms of solution quality.

The structure of the remainder of the paper is as follows. In the next sections, related works on OVRP are presented and then the proposed idea based on ACO called HACO is explained in great detail. In addition, using a new candidate list, building the solution simultaneously by a new transition rule, applying four powerful local search techniques to improve the solution, and proposing a new method for updating global pheromone information which are four main steps of HACO are also described in more detail in the same section. In computational experiments section, the proposed algorithm is compared with some of the other algorithms on standard problems belonging to OVRP library. Finally, some concluding remarks are given.

RELATED WORKS

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 operations research literature. However, several researchers have used some algorithms especially metaheuristic 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.

The OVRP is used to find the best Hamiltonian path for each set of customers assigned to a vehicle. Therefore, this sub-problem is NP-hard because it can be converted into an equivalent Hamiltonian cycle. As a result, the whole problem (e.g. OVRP) belongs to NP-hard problems. Therefore, most of the practical examples of this problem cannot be solved by exact algorithms to optimality within reasonable time and the algorithms used in practice are the heuristic and metaheuristic algorithms. Therefore, research on open vehicle routing problems (VRPs) has recently concentrated on devising effective heuristic and especially metaheuristic algorithms for solving VRPs using a permissible solution instead of the optimal solution. These approaches can obtain feasible solutions within a reasonable computing time and the best of these algorithms can find the optimal or near optimal solutions in an acceptable computing time. In other words, heuristic methods cannot guarantee any specified solution quality. Many of them, however, are known to provide good results in a short time even for large instances.

For example, an efficient tabu search is proposed for OVRP by *Brandão* (2004) in which the neighborhood structure is introduced by insertions and swaps between different routes. Infeasibilities in middle solutions are managed through penalizing the objective function by two penalty terms including capacity violation and route length violation. In this problem a more general variant involving both capacity and route-length constraints are considered. Sariklis and Powells (2004) presented a novel algorithm which comprises two phases for symmetric OVRP without considering a maximum route length. This problem is a real problem of express airmail distribution in the USA and contains many applied features such as delivery or pickup time windows and total route length and capacity of the plane. In this paper, the customers are assigned to studied clusters, taking into account the capacity constraint and trying to make the minimum number of clusters in phase I. Then, reassignments of customers to different clusters are done with the aim of decreasing the travelling cost according to some given constraints. In phase II, each cluster is changed into an open route. In this step, the algorithm starts with a minimum spanning tree (MST) and then applies to it a set of operations in which the objective is to convert it into a minimum cost path.

A tabu search algorithm was proposed by Fu *et al.* (2005, 2006). In this algorithm, the initial solution is provided by a 'farthest first heuristic' and exchanges are based on the two-interchange generation mechanism. On the other hand, a combination of vertex reassignment, 2-opt, vertex swap, and 'tails' swap within the same route or between two routes are used simultaneously.

Tarantilis *et al.* (2005) offered a single-parameter metaheuristic method for solving a version of the OVRP. In this variant of the problem, the objective is to minimize the total distance covered without directly attempting to minimize the number of vehicles and imposing an upper limit on route length. Besides, this algorithm exploits a list of threshold values to guide intelligently a local search based on a variety of edge and node exchanges. Li *et al.* (2007) developed a variant of record-to-record travel algorithm for the standard VRP to handle open routes. This algorithm avoided the premature convergence and found very good solutions in a short computing time by conducting a better exploration in the feasible space.

Pisinger & Ropke (2007) offered an effective metaheuristic based on adaptive large neighbourhood search algorithm. In their algorithm, customers can be removed randomly from the solution and then reinserted in the cheapest possible route. Furthermore, various removal and insertion heuristics can be used to diversify and intensify the search for better results. A simulated annealing framework is applied to move from one solution to the next. Moreover, several famous metaheuristic algorithms have been proposed for the version involving only capacity constraints. For example, Tarantilis *et al.* (2004b) presented a population-based heuristic and a heuristic of the threshold-accepting type.

Levy (2005) has observed that if a company is not paying after the last delivery, then it needs an efficient path that is not concerned with returning to the depot. In fact, if the compensation model includes mileage, the company wants a path that is not influenced by returning to the depot because that would add extra mileage to the compensation model. This algorithm has the unique capability of finding very good solutions in a short computing time through reducing the size of the neighborhood by exploring only the most potentially promising moves and avoiding the premature convergence. Besides, he used the OVRP in the newspaper home delivery problem in which a carrier is subcontracted by the newspaper company to make deliveries to homes. In this problem, the newspaper company is only concerned with the path to the last delivery and after that point, a carrier is not compensated.

Bodin *et al.* (1983) defined the OVRP encountered by FedEx in generating open delivery routes for airplanes. In this problem, an airplane starts to move from Memphis, makes deliveries to several cities, and does not come back to Memphis. After that, the airplane rests in the last city on the delivery route and begins its pickups from that city. Fu *et al.* (2005) described two further areas of the OVRP applications involving the planning of train services and a set of school bus routes. In the first problem, train starts or ends at the Channel Tunnel and in the second problem pupils are picked up at various locations and brought to school in the morning. Besides, the routes are reversed to take pupils home in the afternoon. A description of the problem of express airmail distribution in the USA is defined so that there is an open pick-up and delivery VRP with capacity constraints and time windows. These authors describe a heuristic algorithm that was used by FedEx to develop an open route for each airplane. This algorithm is based on a variant of the Clarke and Wright algorithm which is currently used by FedEx. This company uses the OVRP in its Home Delivery service to residential-only customers by FedEx contract couriers with vehicles. In this problem, drivers move to the FedEX depot each morning, load packages, and then make deliveries to residents so that the couriers and vehicles do not return to the depot after their last deliveries.

Letchford *et al.* (2007) proposed an innovative local search metaheuristic which examines wide solution neighborhoods for solving the OVRP. In this problem, two objective configurations are considered in which the first one primarily aims at minimizing the number of routes and secondarily minimizing the routing cost, whereas the second one only aims at minimizing the cost of the generated route set.

Repoussis *et al.* (2010) developed a population-based hybrid metaheuristic algorithm for solving the OVRP in which the basic solution framework of evolutionary algorithms combined with a memory-based trajectory local search technique is utilized. The proposed method manipulates a population of μ individuals using a $(\mu + \lambda)-$evolution strategy (Back *et al.,* 1997). By using arcs extracted from parent individuals, a new intermediate population of λ offspring is formed via mutation at each generation. Besides, the selection and combination of arcs are dictated by a vector of strategy parameters. Finally, the quality of each new offspring is further enhanced via a memory-based trajectory local search technique while an elitist scheme directs the selection of survivors.

An Integer Linear Programming (ILP) technique based on destruct-and-repair paradigm was proposed by Salari *et al.* (2010) for solving OVRP. In the proposed algorithm, an initial solution which is expected to be improved is considered and the method follows a destruct-and-repair paradigm. In this method, customers are randomly removed and repaired by solving an ILP model for finding a new improved solution. It should be noted that the ILP can be expanded and adapted to cover other variants of VRP.

Zachariadis & Kiranoudis (2010) proposed a local search metaheuristic whose moves are statically encoded into static move descriptor (SMD) entities to explore the wide neighborhoods within the reasonable computational effort. When a local search operator is applied to the candidate solution, only a limited solution part is modified. Consequently, to explore the next neighborhood only the tentative moves that refer to this affected solution part have to be evaluated again. The search is effectively conducted by storing the SMD entities in Fibonacci heaps insertions and deletions. To diversify the search, a tabu scheme and a penalization strategy are employed.

Mirhassani & Abolghasemi (2011) proposed a real-value version of particle swarm optimization (PSO) for solving the open vehicle routing problem (OVRP). In this paper, a particular decoding technique is proposed for implementing PSO. Furthermore, a vector of the customer's position is constructed in descending order in the decoding method. Then each customer is assigned to a route by taking into account feasibility conditions. Finally, one-point move has been applied to constructed routes which seem promising to result in a better solution.

THE PROPOSED ALGORITHM

The ant colony optimization (ACO) was inspired by the behavior of real ant colonies in nature as they forage for food and find the most efficient routes from their nests to food sources. As some ants travel, they deposit pheromone on the paths which are then followed by other ants. This natural behavior of ants can be used to explain how they can find the shortest path. The increase in pheromone increases the chance of the next ants selecting the path (Yousefikhoshbakht & Khorram, 2012). During pass time, more ants are able to complete the shorter route, pheromone accumulates faster on shorter paths, and longer paths are less reinforced. Dorigo (1992) used this concept and proposed the ACO to solve the combinational optimization problems. The ACO as a population-based approach has been successfully applied to several NP-hard combinatorial optimization problems such as the vehicle routing problem and communications networks. In this paper, a hybrid efficient ant colony optimization algorithm (HACO) is proposed to solve OVRP. The HACO is based on the rank-based ant system (RAS). Although this algorithm is strongly inspired by Ant System (AS), it achieves performance improvements through the introduction of new mechanisms based on ideas not included in the original AS. RAS algorithm proposed by Bullnheimer *et al.* (1997) is another version of the ACO. RAS improved the AS algorithm through ranking the solutions constructed by ants.

Our HACO modifies the transition rule, local and global pheromone updating rules and adds several local search techniques on the algorithm. Furthermore, the algorithm applied the idea of candidate lists to construct vehicle routes. Candidate lists can concentrate the search on promising nodes to reduce the computational effort and save the time for further iterations. The proposed algorithm mainly consists of the iteration of the following three steps. These lead to avoiding premature convergence and then searching over the subspace.

Step 1: Build the solution independently using a new transition rule and candidate list.

Step 2: Apply several local search techniques like insert, swap and 2-opt heuristics for several best known solutions until now in order to improve them.

Step 3: Update the global pheromone information.

In the following subsections, each step in more detail is described.

Construct solutions

In the HACO, for *n* groups, *m* ants are initially positioned randomly on *n* vertices and each ant of the colony attempts to build a solution in parallel. At each step of an iteration of the algorithm, only one customer is chosen for each ant according to a transition rule. This method is continued until each ant constructs its route. It should be noted that an important benefit of this parallelism is that several solutions are built at the same time and they interchange information during the procedure and use the information from previous iterations. Besides, ants utilize pheromone trail and heuristic information to build feasible solutions in the process of constructing solutions. Like AS, the next node *j* from node *i* in the route is selected by ant *k* among the candidate list J_i^k with *nu* number according to the following transition rule which shows the probability of each city being visited:

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta [\kappa_{ij}(t)]^\gamma}{\sum_{r \in J_i^k} [\tau_{ir}(t)]^\alpha [\eta_{ir}(t)]^\beta [\kappa_{ir}(t)]^\gamma} \quad \forall j \in J_i^k \tag{1}$$

Where $\tau_{ij}(t)$ is the amount of pheromone on the edge joining nodes *i* and *j*, $\eta_{ij}(t)$is the heuristic information for the ant visibility measure (e.g., defined as the reciprocal of the distance between node *i* and node *j* for the TSP) and α, β, γ are control parameters. Besides, κ_{ij} is defined as 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$\kappa_{ij} = c_{i0} + c_{0j} - c_{ij}$ where node 0 is the depot andc_{ij} denotes the distance between nodes *i* and *j*.

A candidate list is used to determine the next customer selected in a vehicle route and is one of the best methods for improving the quality of the solutions and computational time in the vehicle routing problems (Bullnheimer *et al.*, 1998). Each individual customer is devoted a candidate list based on the distance to all other customers in the customer set which is not visited until now.

In this method, the closest customers are only considered in the candidate list for the current customer and are made available for selection as the next customer to be visited in the route. The size of the candidate list has been determined by restricting its size to a fraction of the total number of customers in the problem. For example, previous research set the candidate list size equal to 25% of the total number of customers regardless of the number of customers (Bullnheimer *et al.,* 1998). For problems with fifty customers the candidate list is limited to the rounded integer value of twelve. It is noted that this restriction prevents the algorithm from wasting its efforts to consider moves to customers which are a great distance from the current customer and have very little chance of creating an improved solution to the problem. Moreover, problems with two hundred customers are common in different versions of the VRP and therefore candidate lists might include as many as fifty customers used in a previous research (Bullnheimer *et al.,* 1997). As a result, in order to decrease the computational time and increase the probability of obtaining a higher-quality solution, upper and lower limits [a,b] are fixed to the number of candidate list *nu*. If the size of this list is not within the minimum *a* and the maximum *b*, then *a* or *b* is allocated to it according to formula (2).

$$nu = \begin{cases} a & n/4 < a \\ b & n/4 > b \\ int(n/4) & otherwise \end{cases} \tag{2}$$

Local search

Most successful metaheuristic methods have paid attention to global search and search in the whole solution space as far as possible. As the algorithm proceeds, it moves to better solutions and the global search switches to a local search. We have also factored in this issue and several local search techniques have been used to improve the solution further. A local search starts with an initial solution and searches within neighborhoods for finding better solutions. In the proposed algorithm, after all the ants have constructed their solutions, the σbest solutions found until now are improved by applying several local searches including insert exchange on each Hamiltonian path and various paths, and swap exchange and 2-opt local search for two Hamiltonian paths. It should be noted that because a local search is a time-consuming procedure, we only apply these local searches to the σ best solutions up to now, which have not been able to improve yet. The idea here is that a better solution may have a better chance to find a global optimum. In order to achieve this goal, several local search techniques including insert exchange on a route and various routes, swap exchange and 2-opt move with the probability ν, $o, \varpi\, and\, \theta$ respectively are used so that $\nu + o + \varpi + \theta = 1$.

In insert exchange, a node from its position in one route is moved to another position in either the same or a different route. Consequently, while the initial tour is $(0, \ldots, i, i+1, \ldots, j-2, j-1, j, j+1, \ldots, 0)$, the improved one is constructed as $(0, \ldots, i, j, i+1, \ldots, j-2, j-1, j+1, \ldots, 0)$. The move is allowed when it is considered favorable for the performance of the entire algorithm in terms of objective and constraints. This move is demonstrated in Figure 1 (a and b).

Besides, in the swap algorithm two nodes from different routes are exchanged. Consequently, if it is supposed that the initial tour consists of the set of nodes $(0, \ldots, i-1, i, i+1, \ldots, j-1, j, j+1, \ldots, 0)\}$, the improved one is constructed as $(0, \ldots, i-1, j, i+1, \ldots, j-1, i, j+1, \ldots, 0)$. The same procedure is conducted in the case of multiple routes. The move is allowed when it is considered favorable for the performance of the entire algorithm in terms of objective and constraints. This move is demonstrated in Figure 1 (c). The most commonly encountered move is the 2-Opt. In multiple routes, edges $(i, i+1)$ and $(j, j+1)$ belong to different routes, but they form a criss-cross again. The 2-Opt move is applied exactly in the same way as is the case in multiple routes. The move is allowed when it is considered favorable for the performance of the entire algorithm in terms of objectives and constraints. This move is demonstrated in Figure 1 (d).

Note that although 2-opt local search as a powerful global search algorithm is more used at the beginning of algorithm, for global search, insert and swap exchanges are more applied at the end of the algorithm because these algorithms might lead to premature convergence to suboptimal regions. In other words, before the algorithm finishes a complete global search, it tends to adopt a local search technique and consequently relatively weak results are attained. Therefore, whenever the algorithm continues, the probability of θ decreases and the probability of ν, o and ϖ increases. Adding this behavior to the imperialist algorithm's revolution policy leads to creating the proper conditions for the algorithm to escape from local peaks. Thus, as mentioned before, the probabilities of using the 2-opt, insert and swap exchanges at the first step of the proposed algorithm are considered ν, $o, \varpi = 0.20$ and $\theta = 0.40$ and then during the steps of the proposed algorithm, they are gradually converted to ν, $o, \varpi = 0.30$ and $\theta = 0.10$.

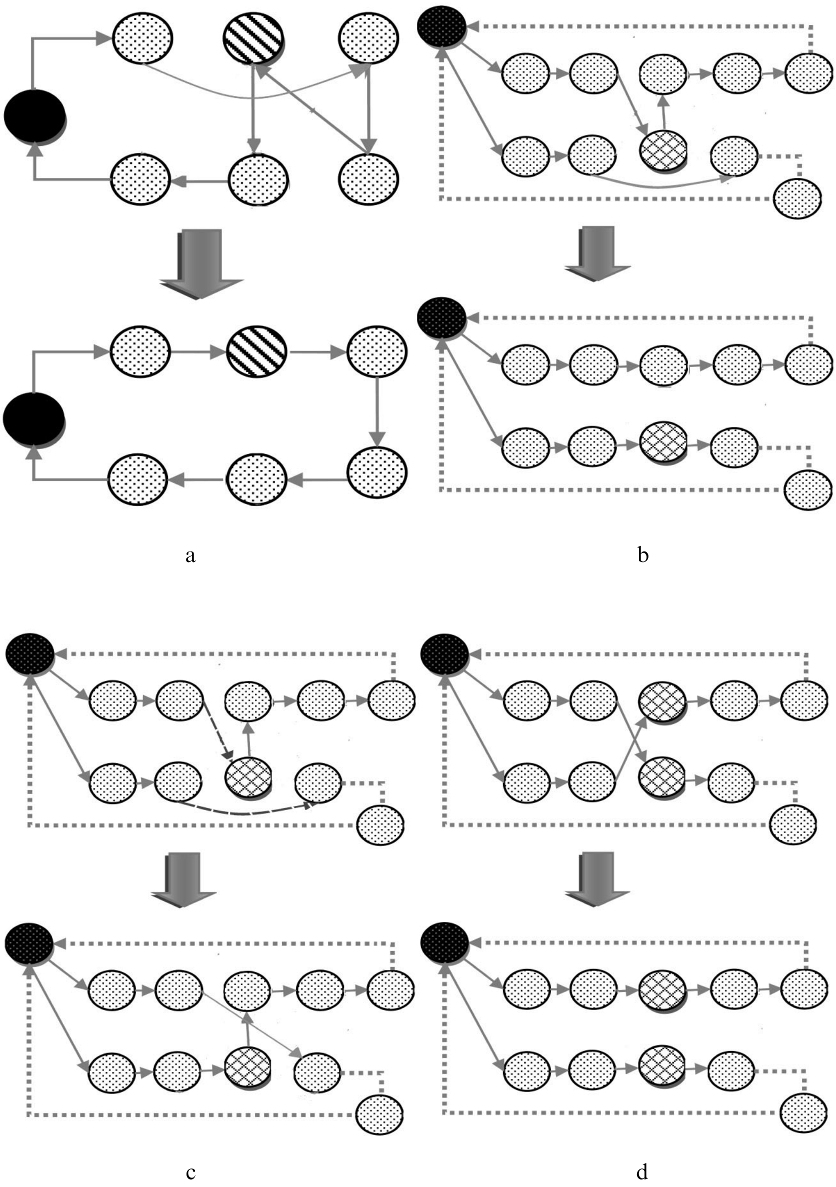

Fig. 1. Insert (a, b), swap (c) and 2-opt exchanges (d)

Updating pheromone

In contrast to AS, the pheromone of all edges belonging to the route chosen by ants is not updated in RAS. The pheromone updating of RAS includes only global updating rules. The pheromone updating formula was meant 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. It results in the new pheromone trail being a weighted average between the old pheromone value and the amount of pheromone deposited. What distinguishes the HACO from the other algorithms is the fact that the amount of releasing pheromone is based on the rank of the best known solution until now. In other words, when all ants have completed their tours and have produced a solution, these steps are repeated for the next group of ants. After producing n solution for the problem in the current iteration, the pheromone level on the edges of the σ best known solution found up to now is updated by formula (3). The rankings of σ elitist ants are updated by comparing the present elitist ant solutions with the current ant solution. If the current found ant solution is better than the μ th elitist ant solution, the current ant solution replaces that solution and becomes the newμ th elitist ant solution. Therefore, the ranks of the previous μ th elitist ant solution and elitist ant solutions below it are increased by one. In order to keep σ elitist ants distinct from together, the current solution is only considered, if the total tour length of the current solution is different from the total tour length in any of the elitist ant solutions.

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

Where:

$$\Delta\tau_{ij}^{\mu}(t) = \begin{cases} (\sigma - \mu + 1).QL^{\mu}(t) & (i,j) \in S^{\mu} \\ 0 & (i,j) \notin S^{\mu} \end{cases} \tag{4}$$

μ : The variable indicating ranking index.

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

S^{μ} : The edges traversed by an ant which has gained the μ th rank in finding the best solution.

σ : The number of solutions which have been ranked.

Q : A constant coefficient determined by the user.

$L^{\mu}(t)$: The length of the path traversed by the μth ant.

COMPUTATIONAL EXPERIMENTS

In this section, solution values and running times of the proposed algorithm on the test problems are presented. We first set the parameters which we use in this algorithm and then the computational results are described in more detail.

Parameter sensitivity

There are many parameters in the HACO algorithm, like every metaheuristic algorithm and the values of these parameters affect directly or indirectly the final solution quality. As a result, a parameter setting procedure is necessary to reach the best balance between the quality of the solutions obtained and the required computational attempt. The goal is to find some robust parameters which allow an algorithm to find high quality solutions for a wide range of problem instances with different features. Because there is no way of defining the most effective values of the parameters, a selection of some of the best parameters for solving the OVRP are considered and finally the best one is selected. For achieving this goal, C1 as a problem instance and several values for each parameter are tested while all the others are held constant. We understand that the most influential parameters in the proposed algorithm, which directly affect the quality of the final solution are: the used number of ants (m), the power parameter of the amount of pheromone on the edge (i,j) (α), the power parameter of the amount of ant visibility value (β), the power parameter of the amount of saving value (γ), a constant coefficient in the formula (3) (Q), minimum number of candidate list (a), maximum number of candidate list (b), the evaporation rate of pheromones (ρ), a number of ants which have been ranked and global updating pheromone has been deposited on their edges (σ), and the termination condition of finishing the algorithm (t).

For determining these parameters, each one for C1 instance was run 10 times while all the others were held constant, and the ones which were selected produced the best computational results concerning the quality of the solution needed to achieve this result. The ranges of six parameters and all of the parameter values have been presented in Table 1. To determine the value of parameters, several alternative values for each parameter were tested. It should be noted that although the results confirm that our parameter setting worked well, it is also possible that better solutions could exist.

Based on the results presented in Table 1, the algorithm with the smaller weight parameter (α) of pheromone trails possesses higher performance. If it is assumed that the initial pheromone trails are large values and if the large control factor of pheromone trail is used, the effect of visibility value is weakened and a premature convergence occurs. Besides, the qualities of the solutions of the algorithms with $\gamma, \beta = 3$ are better than 1,2, 3 and 5. From the test results, it can be understood that by setting the evaporation factor to 0.1, HACO can obtain the best solutions. It can be concluded that if pheromone evaporation is too rapid, the search can be easily trapped in local minima. In other words, a smaller evaporation factor can ensure the sufficient diversity of search space and can guide following ants to explore better solutions. Clearly, increasing the number of iterations of the best solution for termination function (t) devoted to solving the problem should ultimately (after ‘enough’ iterations) improve solution quality. It is noted that the number of iterations assigned to solving the problem has been proved to be important for the solution quality, which can be obtained by the algorithm. If the number of problematic iterations is too small, the problems will be solved unsatisfactorily and the algorithm may converge to a poor solution. If the number of problematic iterations is too large, the algorithm will waste resources and converge too slowly. The settings are chosen to provide a good compromise with respect to this trade-off and have proved to work well independently of the problem instance, which has been solved. However, as discussed in this section, the number of iterations devoted to solving the problem should be kept small to find solutions within reasonable time, in particular for problems of real world size. Therefore, there is a tradeoff between solution quality found at the end of the run time and solution quality found over a short time. Therefore, the best value found for *t* is 15.

Table 1: Parameter setting

Parameters	**Candidate Value**	**The best value**
m	$n, n/2, n/3, n/4, n/5$	n
α	1, 2, 3, 4, 5	1
β	1, 2, 3, 4, 5	3
γ	1, 2, 3, 4, 5	3
Q	50, 100, 200, 300	100
a	3, 5, 7, 9, 11, 13	5
b	17, 19, 21, 23, 25	21
ρ	0.1, 0.2, 0.3, 0.4. 0.5	0.1
σ	3, 6, 9, 12, 15	12
t	3, 5, 10, 15, 20	15

Computational results

The HACO 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 proposed approach is a metaheuristic algorithm, the results are reported for ten independent runs and the best solution found for all instances is reported. Our proposed metaheuristic algorithm was tested on two small and large size sets of OVRP benchmark problems. The first set consisted of fifteen tests numbered from A-n19-k2 to A-n72-k4 with sizes from 19 to 72 nodes. Table 2 shows the result of applying the proposed algorithm compared to three versions of particle swarm optimization (PSO) on problems. In this table, the first column includes the instance name, the second column shows the number of vertices n, and the third column presents the number of used vehicles K, which for all of these instances is fixed at the minimum possible. In other words, the value of *K* has been estimated through dividing the sum of all customer demands by vehicle capacity. It should be noted that these instances do not have the maximum route length restriction. The fourth, fifth and sixth columns of Table 2 are PSO, PSO without one-point move (PSOWO) and PSO without neighbor (PSOWN), the problem name described in (MirHassani & Abolghasemi, 2011). The results of the proposed algorithm and its CPU time are in the seventh and eighth columns and the last column includes the optimal values of these instances obtained from http://www.hha.dk/∼lys/.

Table 2. Comparison results and optimal solutions

Instance	n	K	PSO	PSOWO	PSOWN	HACO		BKS
						Cost	Time	
A-n32-k5	32	5	487.306	940.845	487.306	487.306	7.05	487.306
A-n33-k5	33	5	424.543	749.387	424.543	424.543	7.58	424.543
A-n33-k5	33	6	462.433	799.005	462.433	462.433	9.27	462.433
A-n34-k5	34	5	508.516	838.539	511.088	508.173	6.89	508.173
A-n36-k5	36	5	519.455	947.025	526.919	519.455	8.71	519.455
A-n37-k5	37	5	486.243	792.794	486.243	486.243	8.18	486.243
A-n45-k4	45	4	463.896	1249.250	463.896	463.896	11.70	463.896
A-n72-k4	72	4	177.453	643.618	220.813	177.298	28.49	176.999
A-n19-k2	19	2	168.569	197.076	168.569	168.569	5.48	168.569
A-n20-k2	20	2	170.278	218.370	170.790	170.278	4.99	170.278
A-n21-k2	21	2	163.877	229.981	163.877	163.877	5.67	163.877
A-n22-k2	22	2	167.191	204.112	167.191	167.191	5.12	167.191
A-n40-k5	40	5	349.552	568.116	349.552	349.552	9.86	349.552
A-n45-k5	45	5	391.809	637.533	401.531	391.809	10.84	391.809
A-n50-k7	50	7	407.727	720.973	416.938	397.376	11.59	397.376

Table 2 shows that PSO in all examples except for three instances including A-n34-k5, A-n72-k4 and A-n50-k7 has been able to find equal values for solutions compared with HACO. However, HACO has found optimal solutions for all the instances except for A-n72-k4 in 93% . Generally, the results show that the PSOWO and PSOWN have had a weak performance, but the results of

PSOWN are much better than PSOWO. Besides, the proposed algorithm has been able to improve the performance of the PSOWN and can obtain better results for five instances compared to PSOWN. As a result, the HACO is the best algorithm and the performances from the worst to the best belong to PSOWO, PSOWN, PSO and HACO.

We now consider the large-scale vehicle routing problems in which the number of customers ranges in size from 50 to 199. Each problem exhibits a geometric symmetry, which allows us to visually estimate a solution. In all of these instances, the vertices are taken to be points located in the Euclidean plane. The cost of an edge is then taken to be equal to the Euclidean distance between its end-vertices computed with real numbers. There are 16 test problems identified by their original number and prefixed respectively with the letters C and F available in the literature, and they are summarized in Table 3. The fourteen problems named C1-C14 are taken from Christofides *et al.* (1979), and two problems represented by F11-F12 are taken from (Fisher & Jaikumar, 1978). Furthermore, 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. All problems are available online (see www.branchandcut.org/VRP/data/ and http://people.brunel.ac.uk/mastjjb/jeb/info.html).

In Table 3, some of the characteristics of these problems are described. The first column includes the instance name, the second and third columns show the number of customers n, and the number of used vehicles K, which is fixed at the minimum possible for all of these instances. In other words, the value of *K* has been estimated through dividing the sum of all customer demands by vehicle capacity. Besides, the fourth column which shows the value of *L*denotes the maximum route length. Seven of the problems have a route-length restriction. The objective of the computational experiments is to compare the performance of the HACO with several famous metaheuristic algorithms. For achieving this goal, seven different metaheuristic approaches given in the literature for the OVRP such as tabu search by Fu *et al.* (2005, 2006) (TSF) and *Brandão* (2004) (TSB, TSAN), record-to-record travel algorithm (ORTA) by Li *et al.* (2007), variable neighborhood Search (VNS) by Fleszar *et al.* (2009), threshold accepting approach by Tarantilis *et al.* (2004b) (BATA) and adaptive large neighborhood search (ALNS) by Pisinger & Ropke (2007) were considered. Moreover, TSB is proposed by *Brandão* (2004), TSR is proposed by by Fu *et al.* (2005), BR is proposed by by Tarantilis *et al.* (2005) and LBTA is proposed by Tarantilis *et al.* (2004a). In this table, two sub-columns which include the best gained solution and CPU time are allocated to each algorithm. Furthermore, the best algorithms gaining minimum vehicles with the least distance and the best algorithms gaining minimum distance with the least number of vehicles for all of the instances are presented. Finally, the last column shows best known solution (BKS) by the various algorithms until now. All the times in the tables are in seconds. For each problem, the proposed algorithm used the minimum number of vehicles as specified by the lower bound of K. It should be 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. Since papers on the OVRP tend to report results only for the floating point versions, in this paper we do the same. This table indicates that some algorithms used different number of vehicles shown in the brackets.

To measure the efficiency and the quality of an algorithm, a simple criterion is to compute the number of optimal solutions found in specific benchmark instances by algorithm. As it can be seen from Table 3, the proposed algorithm HACO finds the optimal solution for 7 out of 16 problem instances in a reasonable time and these solutions have been published in the literature. Moreover, TSB, TSF, TSAN, BATA, ORTR, ALNS and VNS have been able to find 0,0,0,4,5,4 and 6 optimal solutions from among these instances. These results indicate that HACO is a competitive approach compared to mentioned algorithms and the results are much better than the ones found by these algorithms. As it is shown in (Tarantilis *et al.* 2008), direct comparisons of the required computational times cannot be conducted as they closely depend on various factors such as the processing power of the computers, the programming languages, the coding abilities of the programmers, the compilers and the running processes on the computers.

Table 3. Results of the HACO compared to other metaheuristic algorithms

Instance	n	K	L	TSF		TSB		TSAN		ORTR		Minimum distance with least number of vehicles		BKS
				Cost	Time	Cost	Time	Cost	Time	Cost	Time			
C1	50	5	-	416.1	0.8	416.1	0.00	438.2	1.7	416.06	6.2	408.5	ALNS, HACO	408.5
C2	75	10	-	567.1	7.8	574.5	1.30	584.7	4.9	567.14	31.3	564.06 [11]	BR, BATA, LBTA, TSR, ALNS	564.06 [11]
C3	100	8	-	643.1	3.0	641.6	0.29	643.4	12.3	639.74	39.5	617	ALNS, TSR, HACO	617
C4	150	12	-	738.9	6.8	740.8	1.05	767.4	33.2	733.13	128.6	733.13	VNS, ALNS, ORTR	733.13
C5	199	16	-	879.0 [17]	61.9	953.4	6.40	1010.9	116.9	924.96	380.6	870.26 [17]	ALNS, LBTA	870.26 [17]
C6	50	5	180	413.0	0.6	412.96	0.00	416.0	1.4	412.96	10.3	400.6 [6]	ALNS, TSF	400.6 [6]
C7	75	10	144	568.5 [11]	6.3	634.54	8.81	581.0 [11]	3.4	568.49 [11]	32.2	560.4 [11]	ALNS, TSR, HACO	560.4 [11]
C8	100	8	207	647.3	10.7	644.63	0.00	652.1	10.4	644.63	53.2	638.2 [9]	ALNS	638.2 [9]
C9	150	12	180	761.3 [14]	46.6	785.2	3.61	827.6 [14]	25.2	756.38 [14]	195.1	752.0 [14]	ALNS	752.0 [14]
C10	199	16	180	903.1	51.9	884.63	1.02	946.8	100.1	876.02	363.5	875.67 [17]	ALNS, VNS	875.67 [17]
C11	120	7	-	724.5	27.5	683.4	0.19	713.3	15.7	682.54	121.6	679.60 [11]	BATA, ALNS	679.60 [9]
C12	100	10	-	534.7	4.2	535.1	0.16	543.2	7.8	534.24	32.9	534.24	BR, ORTR,ALNS, VNS, BATA, HACO	534.24
C13	120	7	648	922.3 [12]	9.6	943.66	4.38	994.3	25.8	896.50 [12]	120.3	896.50 [12]	ORTR, HACO, ALNS	896.50 [12]
C14	100	10	936	600.7	2.5	597.3	0.92	651.9 [12]	8.1	591.87	62.9	591.87 [11]	ORTR, ALNS, VNS	591.87 [11]
F11	71	4	-	177.0	0.4	177.0	0.23	179.5	5.7	177.00	19.5	175.0	ALNS, HACO	175.0
F12	134	7	-	778.6	2.8	781.2	1.50	825.9	32.7	769.66	158.2	769.66	VNS, ALNS, ORTR, HACO	769.66

Table 3. Results of the HACO compared to other metaheuristic algorithms

Instance	n	K	L	TSF		TSB		TSAN		ORTR		Minimum distance with least number of vehicles		BKS
				Cost	Time	Cost	Time	Cost	Time	Cost	Time			
C1	50	5	-	416.06	17.6	412.96 [6]	38.62	408.5	230	408.5	10.29	408.5	HACO	408.5
C2	75	10	-	567.14	29.0	564.06 [11]	68.89	564.06 [11]	530	568.45	24.12	567.14	VNS, ORTR	564.06 [11]
C3	100	8	-	639.74	239.6	642.42	56.54	617	1280	617	44.51	617	ALNS, TSR, HACO	617
C4	150	12	-	733.13	585.0	736.89	81.69	733.13	2790	740.51	95.39	733.13	VNS, ALNS, ORTR	733.13
C5	199	16	-	905.96	292.1	879.37	98.13	870.26 [17]	2370	879.37	241.09	879.37	BATA, HACO	870.26 [17]
C6	50	5	180	412.96	75.8	-	-	400.6 [6]	310	413.00	11.18	400.6 [6]	ALNS, TSF	400.6 [6]
C7	75	10	144	596.47	22.3	-	-	560.4 [11]	330	560.4 [11]	30.51	583.19	ALNS	560.4 [11]
C8	100	8	207	644.63	587.6	-	-	638.2 [9]	1140	645.16	43.61	638.2 [9]	ALNS	638.2 [9]
C9	150	12	180	760.06	1094.1	-	-	752.0 [14]	1850	757.92	82.82	757.84 [13]	ALNS	752.0 [14]
C10	199	16	180	875.67	1252.4	-	-	875.67 [17]	2240	903.10	193.72	875.67 [17]	ALNS, VNS	875.67 [17]
C11	120	7	-	682.12	231.6	679.60 [9]	37.67	679.60 [9]	1410	682.12	43.88	682.12	VNS, ALNS, HACO	679.60 [9]
C12	100	10	-	534.24	163.7	534.24	84.54	534.24	1180	534.24	45.76	534.24	BR, ORTR,ALNS, VNS, BATA, HACO	534.24
C13	120	7	648	904.04	1820.1	-	-	896.50 [12]	1160	896.50 [12]	56.83	909.80 [11]	ALNS	896.50 [12]
C14	100	10	936	591.87	389.0	-	-	591.87 [11]	750	602.38	41.04	591.87 [11]	ORTR, ALNS, VNS	591.87 [11]
F11	71	4	-	178.09	140.2	-	-	175.0	1040	175.00	32.73	175.0	ALNS, HACO	175.0
F12	134	7	-	769.66	1237.5	-	-	769.66	3590	769.66	81.86	769.66	VNS, ALNS, ORTR, HACO	769.66

A simple criterion to measure the efficiency and the quality of an algorithm is to compute the relative average of percentage deviation (PD) of its solution from the BKS on specific benchmark instances. Figure 2 shows the percentage deviation (Gap) of our algorithm and the ones for seven other metaheuristic algorithms. Gap is computed by formula (5) where$c(s^{**})$ is the best solution found by each algorithm for a given instance, and $c(s^{*})$is the overall BKS for the same instance on the Web. From Figure 2 we conclude that the proposed method has the best deviation from the BKS. In more detail, the best algorithm is HACO which has found the best known solutions for all 7 examples including C1, C3, C7, C9, C13, F11 and F12 and is competitive with other algorithms. However, in other instances, the proposed algorithm finds nearly the BKS, i.e. the gap is about as high as 1. On the whole, the average Gap for TSF, TSB, TSAN, ORTA, VNS, BATA, ALNS, and HACO respectively are 2.01, 3.16, 6.34, 1.26, 1.55, 0.97, 0.82 and 1.34. The performance Comparison of results shows that the proposed HACO clearly yields better solutions than the other algorithms.

$$Gap = c(s^{**}) - c(s^{*})c(s^{*}) \times 100 \tag{5}$$

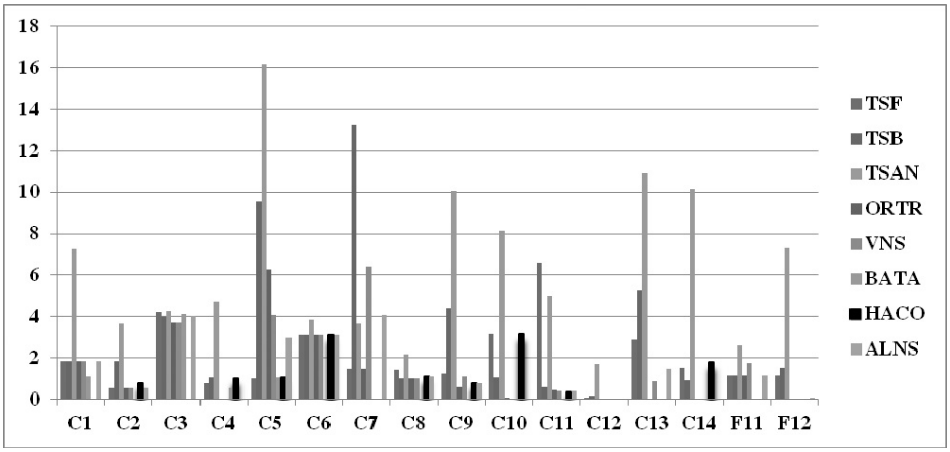

Fig. 2. Comparison Gap of the metaheuristic algorithms

CONCLUSION

In open vehicle routing problem (OVRP) the vehicles do not return to the depot after delivering the packages to the last customer. Although the practical importance of this problem was established some decades ago, it has received very little attention from researchers. An effective hybrid efficient ACO called HACO for the OVRP has been proposed. In addition to introducing some modifications to improve the algorithm, we compared its performance with other metaheuristic algorithms which were designed for the same purpose and have been published recently. Computational results on thirty one benchmark problem instances demonstrated the competitiveness and the accuracy of the proposed method with fixed configuration of parameters and reasonable computational burdens. A research direction worth pursuing will be towards the investigation of more powerful local search techniques which will incorporate intelligent pattern-identification mechanisms. Besides, we are convinced that this strength can be combined with other metaheuristic approaches in the future works. Moreover, this algorithm will be applied a lot more in the future with all versions of vehicle routing problems.

ACKNOWLEDGEMENT

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

REFERENCES

Back, T., Hammel, U. & Schwefel, H. P. 1997. Evolutionary computation: comments on the history and current state. IEEE Trans Evolution Computation, **1**(1): 3-17.

Bodin, L., Golden, B., Assad, A. & Ball, M. 1983. Routing and scheduling of vehicles and crews: The state of the art. Computers and Operations Research, **10** (2): 63-211.

Brandão, J. 2004. A tabu search algorithm for the open vehicle routing problem. European Journal of Operational Research, **157**(3): 552-564.

Bullnheimer, B., Hartl, R. F. & Strauss, C. 1997. A new rank based version of the ant system - A computational study. Central European Journal for Operations Research and Economics, **7**(1), 25-38.

Bullnheimer. B., Hartl, R. F. & Strauss, C. 1998. Applying the ant system to the vehicle routing problem. In: Voss S, Martello S, Osman IH, Roucairol C, editors. Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Boston: Kluwer.

Bullnheimer. B., Hartl, R. F. & Strauss, C. 1999. An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research, 89: 319-328.

Christofides, N., Mingozzi, A. & Toth, P. 1979. The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi L, editors. Combinatorial optimization. Chichester: Wiley; 315-338.

Dorigo, M. 1992. Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 140.

Fisher, M. L. & Jaikumar, R. 1978. A decomposition algorithm for large-scale vehicle routing, Working Paper 78-11-05, Department of Decision Sciences, University of Pennsylvania.

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, **195**: 803-809.

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, **56**: 267-274.

Fu, Z., Eglese, R. & Li, L. Y. O. 2006. Corrigendum: A new tabu search heuristic for the open vehicle routing problem. Journal of the Operational Research Society, **57**: 1018.

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.

Levy, L. 2005. Private communication. RouteSmart Technologies, Inc.

Li, F., Golden, B. & Wasil, E. 2007. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. Computers and Operations Research **34**: 2918-2930.

MirHassani, S. A. & Abolghasemi, N. 2011. A Particle Swarm Optimization Algorithm for Open Vehicle Routing Problem. Expert Systems with Applications, **38**(9): 11547-11551.

Pisinger, D. & Ropke, S. A. 2007. General heuristic for vehicle routing problems. Computers and Operations Research, **34**: 2403- 2435.

Repoussis, P. P., Tarantilis, C. D., Brãysy, O. & Ioannou, G. 2010. A Hybrid Evolution Strategy for the Open Vehicle Routing Problem. Computers & Operations Research, **37** (3): 443-455.

Saadati Eskandari, Z. & Yousefikhoshbakht, M. 2012. Solving the Vehicle Routing Problem by using an Effective Reactive Bone Route Algorithm. Transportation Research Journal, **1**: 51-67

Salari, M., Toth, P. & Tramontani, A. 2010. An ILP improvement procedure for the open vehicle routing problem. Computers and Operations Research, **12** (2): 2106-2102.

Sariklis, D. & Powell, S. 2000. A heuristic method for the open vehicle routing problem. Journal of the Operational Research Society, **51**: 564-573.

Syslo, M., Deo, N. & Kowaklik, J. 1983. Discrete Optimization Algorithms with Pascal Programs. Prentice Hall, Inc.

Schrage, L. 1981. Formulation and structure of more complex/realistic routing and scheduling problems. Networks, **11**: 229-232.

Tarantilis, C. D., Zachariadis, E. & Kiranoudis, C. 2008. A guided Tabu search for the heterogeneous vehicle routing problem. Journal of the Operational Research Society, **59**, 1659-1673.

Tarantilis, C. D., Ioannou, G., Kiranoudis, C. T. & Prastacos, G. P. 2005. Solving the open vehicle routing problem via a single parameter metaheuristic algorithm. Journal of the Operational Research Society, **56**: 588-596.

Tarantilis, C. D., Diakoulaki, D. & Kiranoudis, C. T. 2004a. Combination of geographical information system and efficient routing algorithms for real life distribution operations. European Journal of the Operational Research, **52**: 437-53.

Tarantilis, C. D., Ioannou, G., Kiranoudis, C. T. & Prastacos, G. P. 2004b. A threshold accepting approach to the open vehicle routing problem. RAIRO Operations Research, **38**: 345-360.

Yousefikhoshbakht, M. & Khorram, E. 2012. Solving the Vehicle Routing Problem by a Hybrid Meta-Heuristic Algorithm. Journal of Industrial Engineering International, **8**(11): 1-9.

Yousefikhoshbakht, M. & Sedighpour, M. 2012. A Combination of Sweep Algorithm and Elite Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem. Proceedings of the Romanian Academy A, **13**(4): 295-302.

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.

Submitted : *21/07/2013*
Revised : *22/09/2013*
Accepted : *06/11/2013*

حل مشكلة توجيه المركباتالمفتوحة بواسطة هجين مستعمرة النمل المثلى

محمد صديقهبور[*] و وحيد أحمدي[**] و ماجد يوسفي خوشباكت و فرزاددیدیفار و فرهاد رحمتي[***]

[*] جامعة ازاد الاسلامية - فرع هميدان - هميدان - إيران
sedighpour@iauh.ac.ir
[**] جامعة ازاد الإسلامية - فرع احفاز - احفاز - إيران
[***] قسم الرياضيات وعلم الحاسوب - جامعة أميركبير للتكنولوجيا - طهران - إيران

خلاصة

مشكلة توجيه المركبات المفتوحة هي بديل من مشكلة توجيه المركبات حيث لا يطلب من المركبات للعودة إلى نقطة الانطلاق بعد الانتهاء من الخدمة. بما ان هذه المشكلة ضمن المشاكل صعبة الحل، فقد استخدمت العديد من التقنيات فوق الإرشادية مثل مستعمرة النمل المثلى لإيجاد حل لها في السنوات القليلة الماضية. الأخيرة لديها بعض أوجه القصور مثل بطء سرعة الحوسبة والتقارب محلياً. لذلك، في هذه الورقة نقترح هجين مستعمرة نمل مثلى تدعى هاكو والتي فيها قاعدة جديدة للحالة الانتقالية وقائمة مرشحين أكفاء وعدة أساليب فعالة للبحث المحلي وقاعدة جديدة لتحديث الفرمون تستخدم جميعاً بهدف الحصول على أفضل الحلول. هذا وتظهر التجارب نجاح الخوارزمية في التوصل لحلول تقارب الحلول الافضل بما يفوق عن 97٪ لواحد وثلاثين مسألة تقليدية من مجموعة مسائل قياسية. كما تبين ان خوارزمية هاكو المقترحة قد توصلت إلى أفضل حل لواحد وعشرين مسالة تقليدية، وهي تنافس ثمانية خوارزميات بديلة لحل مشكلة توجيه المركبات المفتوحة. علاوة على ذلك، يشكل حجم قوائم المرشحين المستخدمة من قبل الخوارزمية عامل أساسي في ايجاد حلول محسنة ومدة الحوسبة الخاصة بالخوارزمية تفضل عن مثيلاتها الخاصة بطرق الحل الأخرى.

الكلمات الرئيسية: مستعمرة النمل المثلى؛ قائمة المرشحين؛تقنيات البحث المحلي؛المشاكلصعبة الحل؛ مشكلة توجيه المركبات المفتوحة.