

Yousefikhoshbakht, Majid; Khorram, Esmale

Article

Solving the vehicle routing problem by a hybrid meta-heuristic algorithm

Journal of Industrial Engineering International

Provided in Cooperation with:

Islamic Azad University (IAU), Tehran

Suggested Citation: Yousefikhoshbakht, Majid; Khorram, Esmale (2012) : Solving the vehicle routing problem by a hybrid meta-heuristic algorithm, Journal of Industrial Engineering International, ISSN 2251-712X, Springer, Heidelberg, Vol. 8, pp. 1-9, <https://doi.org/10.1186/2251-712X-8-11>

This Version is available at:

<https://hdl.handle.net/10419/78588>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/2.0/>

ORIGINAL RESEARCH

Open Access

Solving the vehicle routing problem by a hybrid meta-heuristic algorithm

Majid Yousefikhoshbakht^{1*} and Esmale Khorram²

Abstract

The vehicle routing problem (VRP) is one of the most important combinatorial optimization problems that has nowadays received much attention because of its real application in industrial and service problems. The VRP involves routing a fleet of vehicles, each of them visiting a set of nodes such that every node is visited by exactly one vehicle only once. So, the objective is to minimize the total distance traveled by all the vehicles. This paper presents a hybrid two-phase algorithm called sweep algorithm (SW) + ant colony system (ACS) for the classical VRP. At the first stage, the VRP is solved by the SW, and at the second stage, the ACS and 3-opt local search are used for improving the solutions. Extensive computational tests on standard instances from the literature confirm the effectiveness of the presented approach.

Keywords: Ant colony system, NP-hard problems, Sweep algorithm, Vehicle routing problem

Background

One of the parameters that is always in mind in production and services is the reduction of the product's expense since lower product costs yield increase in the company's competitive advantage in terms of production and services, so the company's profit is growth. Furthermore, minimizing the transportation cost is one of the cost reduction methods in which the goods are transported from one place to other places with minimum cost. Therefore, effectively resolving the vehicle routing problem (VRP) is incredibly important in a distribution network of a logistics system.

A lot of research has been carried out in the field of logistics, from the traveling salesman problem to complex dynamic routing problems. Among the prominent problems in distribution and logistics, the vehicle routing problem and its extensions have been widely studied for many years, mainly because of their applications in real world logistics and transportation problems. The VRP has been proven to be a non-deterministic polynomial time problem (Bodin & Golden 1981). This means that the time of the VRP solution grows exponentially with increasing the distribution points.

A large number of techniques in the literature deal with the VRP and its variations such as VRP with pickup and delivery, time windows VRP, stochastic VRP, multi-depot VRP and others. Generally, the techniques used for solving the VRP can be categorized into exact, heuristic and hybrid methods. Although exact algorithms are appropriate for instances with small size, they are not often suitable for real instances owing to the computational time required to obtain an optimal solution. There have been many papers proposing exact algorithms for solving the VRP. These algorithms are based on column generation approach (Choi & Tcha 2007), implicit enumeration algorithm (Ruiz et al. 2004), branch-and-bound method (Valle et al. 2011), etc.

The VRP is an NP-hard problem and difficult to solve by exact methods within acceptable computing times (Lee et al. 2003). In other words, when the problem size of the VRP is increased, the exact methods cannot solve it. So, heuristic methods are necessary to be used for solving instances with large size in a reasonable amount of time. These algorithms can be divided into two main groups including classical heuristics and meta-heuristics. Since heuristic approaches are not so efficient for escaping local optimum values, the obtained solutions by them may have a large disparity compared to the best known or sub-optimal solutions. Some of the well-known heuristic algorithms are gravitational emulation search (Clarke

* Correspondence: khoshbakht@iauh.ac.ir

¹Young Researchers Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran

Full list of author information is available at the end of the article

& Wright 1964), local search (Gillett & Miller 1974) and Lin-Kernighan (Christofides et al. 1979).

A new kind of emerged algorithm basically tries to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space in the last 30 years. These methods are nowadays commonly called *meta-heuristics*. Since the meta-heuristic approaches are very efficient for escaping from local optimum, they are one of the best group algorithms for solving combinatorial optimization problem. A great number of meta-heuristic algorithms have been applied to the VRP, including genetic algorithm (Baker & Ayechev 2003), large neighborhood search (Hong 2012; Pisinger & Ropke 2010), tabu search (TS) (Leung et al. 2011), simulated annealing (Osman 1993), memetic algorithm (Ngueveu et al. 2010), neural networks (Su & Chen 1999) and particle swarm optimization (Ai & Kachitvichyanukul 2009). On the other hand, a limited amount of research addressing the VRP has used the ant colony optimization (ACO), with candidate lists and ranking techniques to improve the ability of a single ACO to solve the VRP (Bullnheimer et al. 1998; Bullnheimer et al. 1999).

Recently, many researchers have found that the employment of hybridization in optimization problems can improve the quality of problem solving in comparison with heuristic and meta-heuristic approaches. Since hybrid algorithms such as genetic algorithms with a local search (Prins 2009; Lima et al. 2004), genetic algorithms with sweep algorithm (SW) and nearest addition method (Wang & Lu 2009), ACO and greedy heuristics (Zhang & Tang 2009), simulated annealing and TS (Lin et al. 2009), neural networks and genetic algorithms (Potvin et al. 1996a; Potvin et al. 1996b), genetic algorithms and ACO (Reimann et al. 2001), etc. have greater ability to find an optimal solution, they have been considered by researchers and scientists in recent years.

The ACO approach is one of the famous meta-heuristic algorithms that simulates the ants' food-hunting behavior and is used for solving combinatorial optimization problems that do not have a known efficient algorithm. This technique, one of the most powerful methods compared to other meta-heuristic algorithms nowadays, was introduced by Dorigo et al. in (Dorigo 1992) (Dorigo et al. 1991; Dorigo 1992). Because a limited amount of research addressing the VRP has used the ACO combined with heuristic algorithms, a hybrid meta-heuristic algorithm based on ant colony system is proposed for the VRP in this paper. In this algorithm, the VRP is solved by the SW firstly, and then the ant colony system (ACS) and 3-opt local search are used for improving the solutions.

In the following parts of this paper, in 'VRP mathematical model', we will explain the formulation of VRP.

Then, at the beginning of 'Methods', we specially explain the SW and ACS, and then, the combination of ACS and SW is extendedly explained in this section. In 'Numerical calculations', the proposed algorithm will be compared with some of the other algorithms on standard problems, which are included in the VRP library. Finally in the last section, the conclusions are presented.

VRP mathematical model

The VRP is one of the applied problems in the real world. This problem consists of a large number of customers in which everyone has a known demand level. These customers must be supplied from node 0 called the depot. Delivery routes which start and finish at the depot are required for vehicles so that all customer demands are satisfied and each customer is visited by just one vehicle. Vehicle capacities are given and limited by Q . So, the VRP can be explained mathematically as follows.

Let $G(V, A)$ be a perfect undirected connected graph with a vertex set $V = \{0, 1, \dots, n\}$ and an edge set $A = \{(i, j) : i, j \in V, i \neq j\}$. If the graph is not perfect, we can replace the lack of any edge with an edge that has an infinite size. Vertex 0 represents the depot; each of the other vertex $i \in V - \{0\}$ is a customer with a non-negative demand q_i and a non-negative distance c_{ij} is associated with each arc $(i, j) \in A$ and represents the travel cost from node i to node j . The cost matrix is symmetric, i.e., for all $i, j \in V$, $c_{ij} = c_{ji}$. The use of the loop arc (i, i) is not allowed and defining $c_{ii} = +\infty$ for all $i \in V$.

So, the solution for the VRP determines a set of delivery routes that satisfies the requirements of distribution points and obtains the minimum total cost for all vehicles. In practice, minimizing the total cost is equivalent to minimizing the total distance traveled by $m > 1$ vehicles.

An example of a single solution consisting of a set of routes constructed for a classical VRP is presented in Figure 1, where $m = 4$ and $n = 12$. So, the solution of this example is 0-1-2-3-4-0; 0-5-6-7-0; 0-8-9-10-0; 0-11-12-0.

For presenting the integer linear programming model for VRP, the variables below are introduced:

n = the number of nodes for each instance.

m = the number of vehicles used for each instance.

$$x_{ij} = \begin{cases} 1 & \text{if the vehicle travels directly from node } i \text{ to node } j \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if the node } i \text{ is visited by the vehicle } k \\ 0 & \text{otherwise.} \end{cases}$$

Hence, one of the integer programming formulations for the VRP can be written as follows:

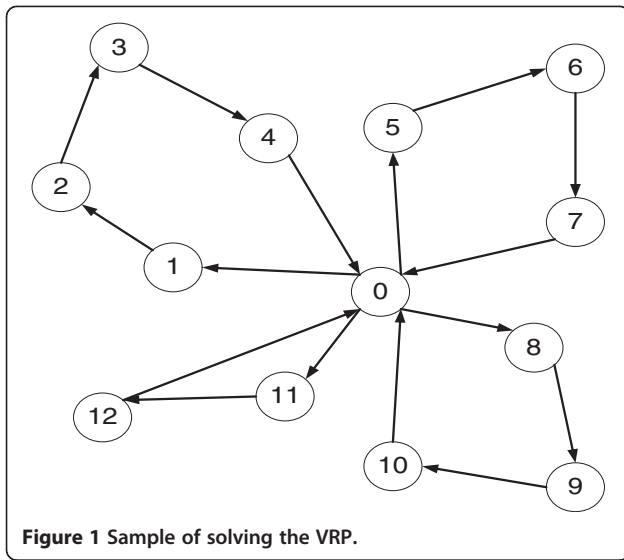


Figure 1 Sample of solving the VRP.

$$\min \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}$$

subject to

$$\sum_{i=0}^n x_{ij} = 1 (j = 1, \dots, n) \quad (1)$$

$$\sum_{i=0}^n x_{i0} = m \quad (2)$$

$$\sum_{j=0}^n x_{ij} = 1 (i = 1, \dots, n) \quad (3)$$

$$\sum_{j=0}^n x_{0j} = m \quad (4)$$

$$\sum_{i=0}^n q_i y_{ik} \leq Q (k = 1, \dots, m) \quad (5)$$

$$\sum_{i \in S} \sum_{j \in N-S} x_{ij} \geq 1 (\phi \neq S \subset \{1, \dots, n\}, |S| \geq 2) \quad (6)$$

$$\sum_{i \in N-S} \sum_{j \in S} x_{ij} \geq 1 (\phi \neq S \subset \{1, \dots, n\}, |S| \geq 2) \quad (7)$$

$$x_{ij} \in \{0, 1\} (i = 0, \dots, n; j = 0, \dots, n) \quad (8)$$

$$y_{ik} \in \{0, 1\} (i = 0, \dots, n; k = 1, \dots, m) \quad (9)$$

The objective function (1) minimizes the total distance traveled in a tour. Constraint sets (2) and (3) ensure that the vehicle arrives once at each node and m times at the

depot. Constraint sets (4) and (5) ensure that the vehicle leaves each node once and leaves the depot m times. Constraint set (6) prevents vehicles from carrying loads more than their capacity. Constraint sets (7) and (8) avoid the presence of sub-tours for each vehicle. Finally, Constraint sets (9) and (10) define binary conditions on the variables.

Results and discussion

Numerical calculations

At the first stage in this section, sensitivity analyses of parameters in the proposed algorithm (PA) are performed, and at the second stage, the SW + ACS, which was discussed in the previous section, is analyzed using several benchmarking problems of Christofides et al. (Christofides et al. 1979) and Taillard (Taillard 1993). The proposed algorithm was coded using MATLAB language executed on a computer with a 1.00-GB RAM and an Intel, 2.80-GHz CPU.

Sensitivity analyses of parameters

In this section, the parameters in our algorithm are tuned, in which there are four parameters including α , β , ρ and q_0 . The ranges of the four parameters were set to $\alpha \in \{1, 2, 3\}$, $\beta \in \{2, 4, 6\}$, $\rho \in \{0.1, 0.2, 0.3\}$ and $q_0 \in \{0.9, 0.95, 0.99\}$. When tuning the parameters, the instance E-n51-k5 was determined as the test problem. Then, the algorithm with each parameter combination for this instance was tested five times.

Based on the gained results, the algorithm with the smaller weight parameter (α) of pheromone trails possesses higher performance. This may be attributed to the fact that in the SW + ACS, the initial pheromone trails are large values. If using the large control factor of the pheromone trail, the effect of visibility value is weakened and results in a premature convergence. In addition, the qualities of the solutions of the algorithms with $\beta = 2$ are better than those of 4 and 6.

From the test results, it can be found that by setting the evaporation factor to 0.1, the proposed algorithm can yield better solutions. This can be attributed to the fact that if pheromone evaporation is too rapid, it more easily results in the search to be trapped in local minimum. In other words, the smaller evaporation factor can ensure the sufficient diversity of search space and guide following ants to explore better solutions.

Finally, the SW + ACS, in which q_0 is set to 0.99, can provide better solutions in comparison with other values. Thus, the combinations of optimal parameters are determined: $\{\alpha = 1, \beta = 2, \rho = 0.1, q_0 = 0.99\}$.

Benchmark instances

In the instances of Christofides et al. (Christofides et al. 1979), which can be downloaded from the DEIS -

Table 1 Comparing algorithms for standard VRP problems

Example	PA	ACO (Mazzeo & Loiseau 2004)	SA + 3-opt	SA (Clark & Wright 1964)	TS (Leung et al. 2011)	BKS (Mazzeo & Loiseau 2004)
E-n51-k5	521	521	578.56	584.64	524.61	521
E-n76-k10	838	877	888.04	900.26	844	832
E-n101-k8	839.2	845	878.70	886.83	835	815
E-n101-k10	823.74	838	824.42	833.51	820	820
E-n121-k7	1050	1,189	1,049.43	1,071.07	1,042.11	1,042.11
M-n151-k12	1,030.46	1,105	1,128.24	1,133.43	1,052	1,028.42
M-n200-k17	1,325.62	1,606	1,336.84	1,395.74	1,378	1,291.45

ACO, ant colony optimization; BKS, best known solution; PA, proposed algorithm; SA, saving algorithm; SA + 3-opt, saving algorithm + 3-opt; TS, tabu search.

Operations Research Group Library of Instances (DEIS - Operations Research Group 2012), the results obtained from calculations by the proposed methods are compared with other algorithms including saving algorithm (SA), saving algorithm + 3-opt (SA + 3-Opt), TS search and ACO. The first two of these algorithms are traditional, and the next two are relatively new. These algorithms were executed on standard instances containing between 51 and 200 nodes of the VRP problem including E-n51-k5, E-n76-k10, E-n101-k8, E-n101-k10, E-n121-k7, M-n151-k12 and M-n200-k17. The number at the right of each instance shows the number of vehicles, and the middle number indicates the number of related customers.

Table 1 summarizes the computational results of the proposed, compared and other mentioned algorithms. Moreover, to show the method's performance more clearly, we present the best known solutions (BKSs) that have been published in the related literature in this table.

Besides, Figure 2 shows a comparison between the gap values of the heuristic and meta-heuristic algorithms, where the gap is defined as the percentage of deviation from the best known solution in the literature. The gap is equal to $100[c(s^{**}) - c(s^*)]/c(s^*)$, where s^{**} is the best solution found by the algorithm for a given instance, and s^* is the overall best known solution for the same

instance on the web. A zero gap indicates that the best known solution is found by the algorithm.

As can be seen from Table 1, the proposed algorithm finds the optimal solution for one out of seven problems that are published in the literature. The results indicate that SW + ACS is a competitive approach compared to the traditional and new meta-heuristics. For instances E-n101-k8 and M-n200-k17, the gap is relatively as high as 3%. However, in other instances, the proposed algorithm finds nearly the best known solution, i.e., the gap is below 0.76%, and overall, the average difference is 1.1%. The performance comparison of results shows that the SW + ACS method clearly yields better solutions than the ACO, saving algorithm and its modified version (SA + 3-opt).

Moreover, computational results of the proposed algorithm and TS show that these algorithms have a close competition, and the proposed algorithm gives four better solutions than the TS. In other words, the performance of the SW + ACS is better in reaching the sub-optimal solution than the TS (Table 1).

From the statistical viewpoint in Table 2, there is no statistical significant difference between the PA and other meta-heuristic algorithms (p value = 0.9881). Furthermore, it can be observed from both the medians plot (Figure 3) and the means plot (Figure 4) that the PA is

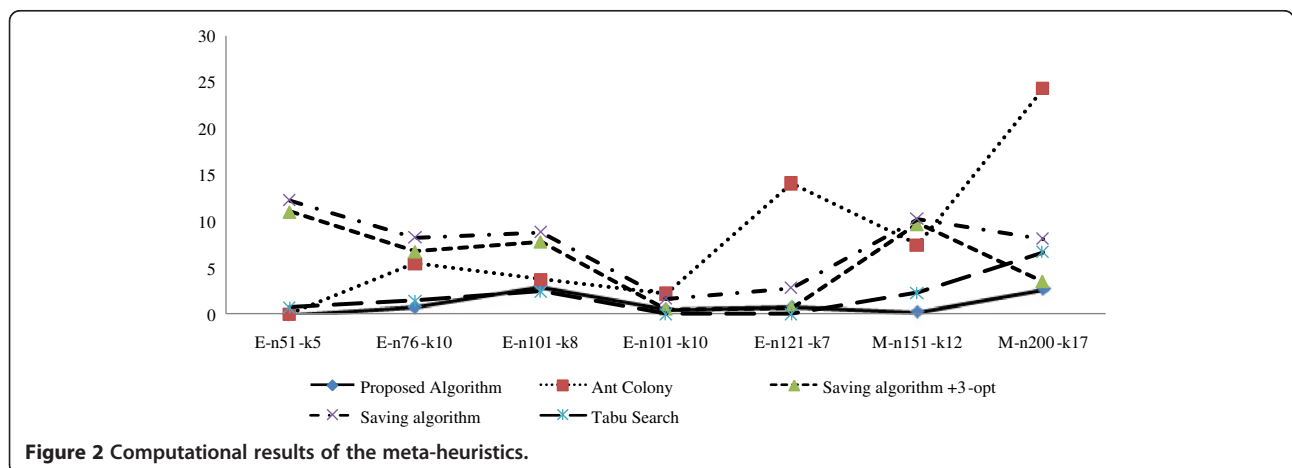


Figure 2 Computational results of the meta-heuristics.

Table 2 ANOVA table

Source	SS	DF	MS	F	The p value
Columns	41,994.6	5	8,398.9	0.12	0.9881
Error	2,607,392.3	36	72,427.6		
Total	2,649,386.9	41			

Df, degrees of freedom; F, F ratio; MS, mean square; SS, sum of the squares.

better than the ACO, SA + 3-Opt and SA algorithms. Also, by observing these figures, we can yield the approximate equality of the PA and TS methods.

In Table 3, the values obtained by the common ACO and proposed algorithm are shown. The results indicate that these two algorithms have very good ability for small problems, have approximately similar behavior and can converge to the best solutions. However, this ability of ACO against the proposed algorithm decreases when the number of nodes and feasible solutions increases.

Table 4 shows the results obtained for the second problem instances including 12 instances proposed by Taillard and presents the comparison between the best results of our algorithm and the BKS. As shown in Table 4, in three cases, the BKS can be found by our algorithm. In other cases, the deviation between the BKS reported and the solution found by the proposed algorithm is very low. In addition, the proposed algorithm finds closely the BKS for most of the instances, i.e., the gap is below 1.10%.

Conclusions

In this paper, a hybrid algorithm combining ACS, SW and 3-opt local search was proposed for solving the VRP. The SW + ACS is more efficient than the ACO.

Especially for large-size problems, this algorithm yields better solutions compared with previous algorithms. Another benefit of this algorithm is that when ACS is used, the number of nodes and complexity of the problem are decreased. As a result, the algorithm can solve these small problems easily. So, it is not necessary for the algorithm to be executed many times for finding the best solutions since the SW and 3-opt local search do not have great time requirements. It also seems that the combination of the proposed algorithm and genetic or simulated annealing algorithms will yield better results. Using the proposed algorithm for other versions of the VRP is suggested for future research.

Methods

Because the VRP is one of the most important combinatorial optimization problems, a hybrid algorithm called SW + ACS is proposed in this paper. In this section, the SW is presented first then the ACS will be explained, and finally, the hybrid algorithm will be analyzed in more detail.

Sweep algorithm

Gillett and Miller (Gillett & Miller 1974) proposed a SW in 1974 for Euclidean networks, which ranks and links demand points by their polar coordinate angle (Figure 5). The polar coordinate angle is calculated as follows:

$$\text{An}(i) = \arctan\{(y(i) - y(0))/(x(i) - x(0))\}$$

It is noted that if $y(i) - y(0) < 0$ and $y(i) - y(0) \geq 0$ then $-\pi < \text{An}(i) < 0$ and $0 \leq \text{An}(i) \leq \pi$, respectively. In this

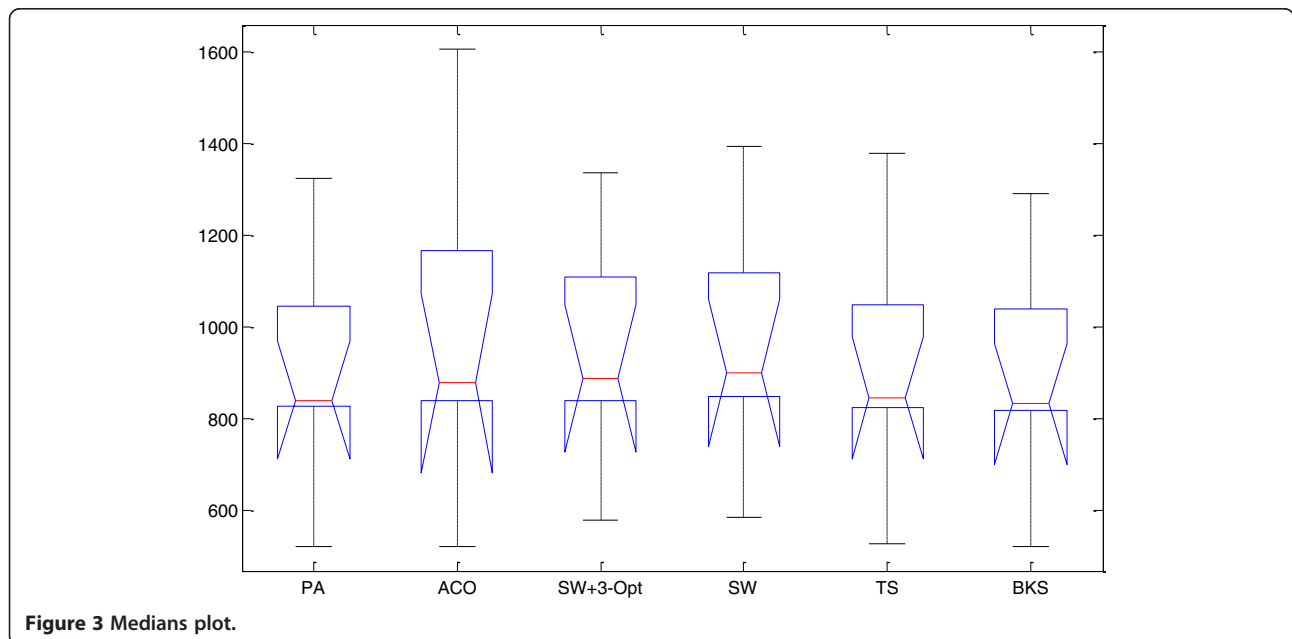


Figure 3 Medians plot.

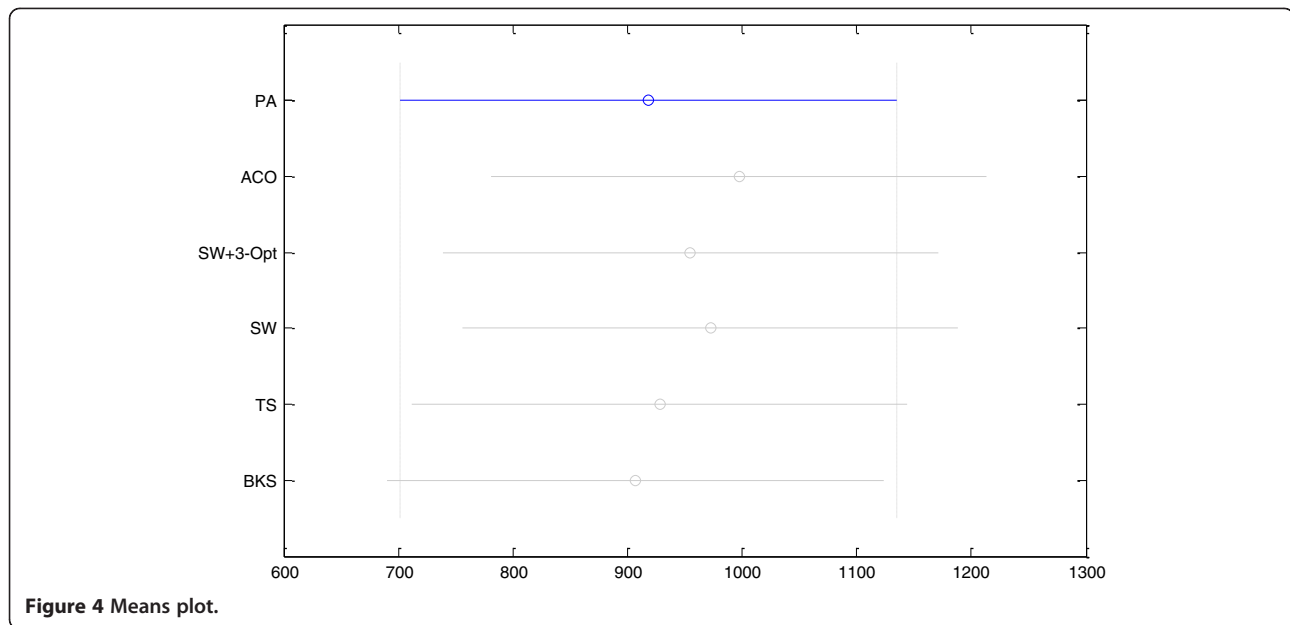


Figure 4 Means plot.

algorithm, which is one of the famous and powerful heuristic algorithms, at first, the polar coordinates of all n customers are calculated, where the depot is node 0 and so $An(1) = 0$.

Then, sweeping (clockwise or counterclockwise) is started from customer i , which has not been visited, from the smallest angle to the largest angle until all customers are included in a tour.

$$0 = An(1) \leq An(2) \leq \dots \leq An(n).$$

Ant colony system

ACS is one of the most powerful versions of ACO. This meta-heuristic algorithm was introduced in 1996 by Dorigo and Gambardella (Dorigo & Gambardella 1997), which was strongly inspired by the ant system (AS). So, this algorithm achieves performance improvements because of the introduction of new mechanisms based on ideas not included in

Table 3 Mean gap values of ACO and the SW + ACS

Example	ACO	PA
E-n51-k5	0	0
E-n76-k10	5.41	0.72
E-n101-k8	3.68	2.97
E-n101-k10	2.20	0.46
E-n121-k7	14.10	0.76
M-n151-k12	7.45	0.20
M-n200-k17	24.36	2.65

ACO, ant colonization optimization; PA, proposed algorithm.

the original AS. The ACS algorithm has two major changes to the rules employed in the AS algorithm, namely:

- (1) A novel transition rule that favors either exploitation or exploration is introduced. Node j that is next to node i , among the unvisited nodes J_i^k , is selected by ant k in the route. According to the following transition rule, the probability of each node being visited is

$$P_{ij}^k(t) = \begin{cases} 1 & \text{if } \{q \leq q_0 \text{ and } j = j^*\} \\ 0 & \text{if } \{q \leq q_0 \text{ and } j \neq j^*\} \\ \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{r \in J_i^k} \tau_{ir}^\alpha(t) \eta_{ir}^\beta(t)} & \text{otherwise} \end{cases} \quad (11)$$

Where

j^* = the unvisited node j in J_i^k for which $[\tau_{ir}(t)][\eta_{ir}(t)]^\beta$ is maximized.

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

$\eta_{ij}(t)$ = the heuristic information for the ant visibility measure defined here as the reciprocal of the distance between node i and node j .

α, β = two control parameters that represent the relative importance of the amount of pheromone on the edge between node i and node j compared to the ant visibility value respectively.

q = a random variable in $[0, 1]$.

q_0 = a given arbitrary parameter fixed before the program is started such that when q is less than q_0 , the ant employs exploitation to select node j^* as the

Table 4 Computational results for the benchmark of Taillard

Problem instance	Size	PA	BKS	Gap
Tai75a	75	1,618	1,618	0
Tai75b	75	1,344	1,344	0
Tai75c	75	1,295	1,291	0.31
Tai75d	75	1,380	1,365	1.10
Tai100a	100	2,050	2,041	0.44
Tai100b	100	1,940	1,939	0.052
Tai100c	100	1,411	1,406	0.36
Tai100d	100	1,585	1,581	0.25
Tai150a	150	3,055	3,055	0
Tai150b	150	2,732	2,656	2.86
Tai150c	150	2,364	2,341	0.98
Tai150d	150	2,660	2,645	0.57

Values in bold are the three cases where the best known solution can be found by our algorithm. PA, proposed algorithm; BKS, best known solution.

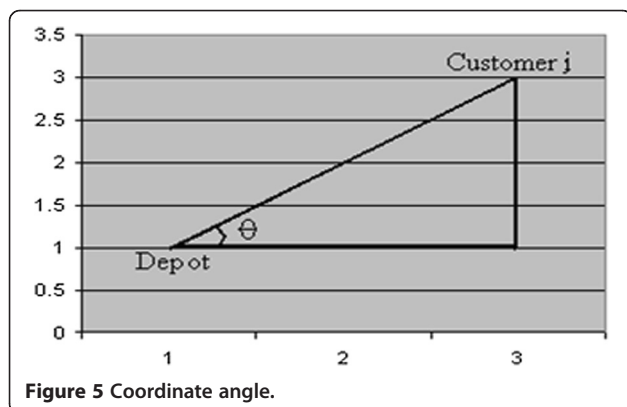
next node in its tour, whereas if q exceeds q_0 , the ant uses probabilistic exploration to select the next node in its tour.

(2) There are two ways for updating pheromone trail as follows:

- Local updating: When an ant moves from node i to node j , it updates the amount of pheromone on the traversed edge using the following formula:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \tau_0 \quad \text{if } (i, j) \in T_k \quad (12)$$

where τ_0 , the initial amount of pheromone, is calculated as $\tau_0 = (nC_i)^{-1}$, n is the number of nodes, C_i is the cost of the initial tour produced by a construction heuristic such as the nearest neighbor



heuristic, SW, etc., and ρ is a parameter called evaporation rate in the range $[0, 1]$ regulating the reduction of pheromone on the edges. It should be noted that local updating has an important effect, such that whenever an ant traverses an edge (i, j) , its pheromone trail τ_{ij} is reduced. So, the edge becomes less desirable for the ants in future iterations. Local updating not only encourages an increase in the exploration of edges that have not been visited yet but also helps avoid poor stagnation situations.

- Global updating: When all tours are generated by ants, the edges that belong to the best tour are updated using the following formula:

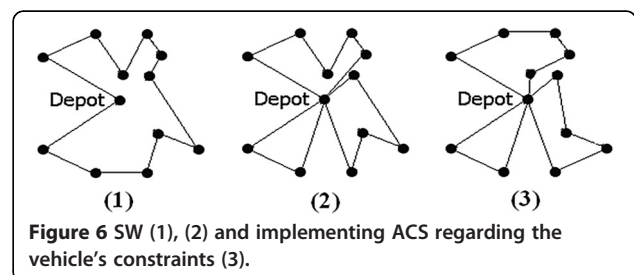
$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho(1/C_b) \quad \text{if } (i, j) \in T_b \quad (13)$$

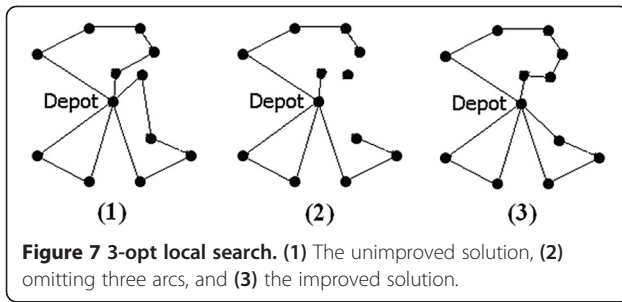
where C_b is the cost of the best tour T_b found yet. It is important to note that the pheromones of the edges belonging to the best tour are only updated in the global updating. This encourages ants to search in the vicinity of this best tour in future iterations.

Hybrid algorithm

From another viewpoint, there are also two heuristic groups for solving combinatorial optimization problems: construction algorithms that produce a feasible solution themselves, and improvement algorithms that can improve the solutions with having a feasible solution. Here, a construction algorithm, namely SW, and two improvement algorithms, namely ACS and 3-opt local search, are used. In this algorithm, the ACS is applied for improving every route of the vehicle; however, the nodes of each vehicle should be unchanged. On the other hand, the 3-opt local search is used for changing nodes and improving each vehicle. In other words, if this algorithm is separated into two parts, the first one to improve the route of the vehicle without changing the vehicle's nodes and the second one to improve it only by changing the vehicle's nodes, then the ACS does operation 1 and the 3-opt local search does operation 2 (Figure 6).

In this method, first, the nodes that should be visited by vehicles are ordered with respect to the depot by SW, and then, they are set in an array. Second, all of the





states in which the above-mentioned order is preserved in spite of a change in their locations are obtained. For example, if 1 2 3 is the order obtained for a feasible solution, then orders 2 3 1 and 3 1 2 are set in other arrays. Then for each array, the vehicle starts to move from the depot and visits the nodes in the arrays in the order described until it is not possible to add a further node. This means that if the load of a vehicle is more than its capacity, it returns to the depot and repeats these steps until there is no more node to be visited. When vehicle

routes are obtained, the ACS is implemented for every route until the best route is obtained.

Furthermore, the vast literature on meta-heuristics tells us that a promising approach for obtaining high-quality solutions is to couple a local search algorithm with a mechanism to generate initial solutions. The ACO's definition includes the possibility of using local search as daemon actions. Daemon actions are used to implement centralized actions that cannot be performed by a single ant. An example of daemon actions is the activation of a local optimization procedure. In implementation, 3-opt local search procedure is used to obtain more improvement in the algorithm's performance.

This algorithm, which is shown in Figure 7, is based on omitting three arcs of the tour that are not adjacent and connecting them again by another method. 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. So, this unique tour will be accepted only if, first, the above constraints are not violated specially regarding each vehicle's capacity,

```

procedure the proposed algorithm
    n is number of nodes;
    m is number of vehicles;
    s* is the random solution;           // s* is the best solution found //
    f* = ∞;
    Set parameters of the proposed algorithm;
    Initialize pheromone trails;
    Use SW to order nodes that should be visited by vehicles with respect to the depot
    and called it R
    gain all of the situations for R and set them to the matrix T
    for i := 1 to length(T) do           // main cycle //
        begin
            Use SW for T(i,:) to produce a solution called T*(i,:)
            For the route of each vehicle in T*(i,:), execute the ACS;
            If new solution is better than T*(i,:), replace it;
            Apply 3-Opt local search algorithm for T*(i,:)
            if f(T*(i,:)) < f* then
                begin
                    f* := f(T*(i,:));
                    s* = T*(i,:);
                end
            end // save the best so far solution //
        end // main cycle //
    show s* and f*
end // procedure //
    
```

Figure 8 Description of the proposed algorithm.

and second, the new tour produces a better value for the problem than the previous solution. Besides, omitting three arcs and reconnecting them are repeated until no improving 3-opt is found.

When all of these n arrays are gained, the best one is selected as the best solution, and the algorithm will be finished. So, it must be noted that although the hybrid algorithm is composed of two heuristic and one meta-heuristic method, the strengths of this algorithm in comparison with other algorithms such as ACO or TS are, first, the less time taken to obtain the solution and, second, the algorithm not needing many iterations to gain a better solution. In other words, when the ACS is activated, the number of nodes is decreased, and the algorithm can reach a good solution during the execution. In addition, because of the lack of division in other meta-heuristic algorithms, the quality of solutions obtained are usually less stable compared to those of this algorithm in various executions. A pseudo-code of the proposed algorithm for the VRP is presented in the Figure 8, while in the current section, the procedures of the algorithm are explained in detail.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MY performed the numerical experiment and carried out sensitivity analysis by taking an illustration. EK managed the study and participated in its design, drafted the manuscript and performed the statistical analysis. Both authors read and approved the final manuscript.

Acknowledgements

We are thankful to the reviewers for their valuable comments.

Author details

¹Young Researchers Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran. ²Faculty of Mathematics and Computer Science, Amirkabir University of Technology, Hafez Avenue, Tehran, Iran.

Received: 27 May 2011 Accepted: 5 March 2012

Published: 7 August 2012

References

- Ai J, Kachitvichyanukul V (2009) Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Comput Ind Eng* 56:380–387
- Baker BM, Ayechew MA (2003) A genetic algorithm for the vehicle routing problem. *Comput Oper Res* 30:787–800
- Bodin L, Golden B (1981) Classification in vehicle routing and scheduling. *Networks* 11:97–108
- Bullnheimer B, Hartl RF, Strauss C (1998) Applying the ant system to the vehicle routing problem. In: Voss S, Martello S, Osman IH, Roucairol C (eds) *Metaheuristics: advances and trends in local search paradigms for optimization*. Kluwer, Boston
- Bullnheimer B, Hartl RF, Strauss C (1999) An improved ant system algorithm for the vehicle routing problem. *Ann Oper Res* 89:319–328
- Choi E, Tcha DW (2007) A column generation approach to the heterogeneous fleet vehicle routing problem. *Comput Oper Res* 34(7):2080–2095
- Christofides N, Mingozzi A, Toth P (1979) *The vehicle routing problem*. Combinatorial optimization. Chichester, Wiley, pp 315–338
- Clark G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Oper Res* 12:568–581

- Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Oper Res* 12:568–581
- DEIS - Operations Research Group (2012) Library of Instances, http://www.or.deis.unibo.it/research_Pages/ORinstances/VRPLIB/VRPLIB.html
- Dorigo M (1992) Optimization, learning and natural algorithms (in Italian), PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, p p 140
- Dorigo M, Gambardella L (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evolutionary Computing*, pp 53–66
- Dorigo M, Maniezzo V, Colomi A (1991) Positive feedback as a search strategy, Technical Report 91–106. Dipartimento di Elettronica, Politecnico di Milano, Italy
- Gillett BE, Miller LR (1974) A heuristic algorithm for the vehicle dispatch problem. *Oper Res* 22:340–349
- Hong L (2012) An improved LNS algorithm for real-time vehicle routing problem with time windows. *Comput Oper Res* 39(2):151–163
- Lee ZJ, Su F, Lee CF (2003) Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. *IEEE Transactions on Systems, Man and Cybernetics, Part B* 33:121–133
- Leung SCH, Zhou X, Zhang D, Zheng J (2011) Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Comput Oper Res* 38(1):205–215
- Lima CMRR, Goldberg MC, Goldberg EFG (2004) A memetic algorithm for the heterogeneous fleet vehicle routing problem. *Electronic Notes in Discrete Mathematics* 18:171–176
- Lin SW, Lee ZJ, Ying KC, Lee CY (2009) Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Syst Appl* 36:1505–1512
- Mazzeo S, Loiseau I (2004) An ant colony algorithm for the capacitated vehicle routing problem. *Electronic Notes in Discrete Mathematics* 18:181–186
- Ngueveu SU, Prins C, Wolfler-Calvo R (2010) An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Comput Oper Res* 37:1877–1885
- Osman LH (1993) Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann Oper Res* 41:421–451
- Pisinger D, Ropke S (2010) Large neighborhood search. In: Gendreau M, Potvin J-Y (eds) *Handbook of metaheuristics of International series in operations research & management science*, 146th edn. Springer, New York, pp 399–419
- Potvin JY, Dube D, Robillard C (1996a) A hybrid approach to vehicle routing using neural networks and genetic algorithms. *Appl Intell* 6:241–252
- Potvin JY, Duhamel C, Guertin F (1996b) A genetic algorithm for vehicle routing with backhauling. *Appl Intell* 6:345–355
- Prins C (2009) Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Eng Appl Artif Intel* 22:916–928
- Reimann M, Shtovba S, Nepomuceno E (2001) A hybrid ACO-GA approach to solve Vehicle Routing Problems. Student Papers of the Complex Systems Summer School, Santa Fe Institute, Budapest
- Ruiz R, Maroto C, Alcaraz J (2004) A decision support system for a real vehicle routing problem. *Eur J Oper Res* 153(3):593–606
- Su CT, Chen HH (1999) Vehicle routing design of physical distribution center. *Journal of the Chinese Institute of Industrial Engineers* 16(3):405–417
- Taillard E (1993) Parallel iterative search methods for vehicle routing problems. *Networks* 23(8):661–673
- Valle CA, Martinez LC, Cunha AS, Mateus GR (2011) Heuristic and exact algorithms for a min–max selective vehicle routing problem. *Comput Oper Res* 38(7):1054–1065
- Wang CH, Lu JZ (2009) A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. *Expert Syst Appl* 36:2921–2936
- Zhang X, Tang L (2009) A new hybrid ant colony optimization algorithm for the vehicle routing problem. *Pattern Recognit Lett* 30:848–855

doi:10.1186/2251-712X-8-11

Cite this article as: Yousefikhoshbakht and Khorram: Solving the vehicle routing problem by a hybrid meta-heuristic algorithm. *Journal of Industrial Engineering International* 2012 **8**:11.