Journal of Computer Science 8 (6): 846-852, 2012
ISSN 1549-3636

Ant Colony Optimization for Capacitated Vehicle Routing Problem

[1]Tan, W.F., [1,2]L.S. Lee, [1,2]Z.A. Majid and [3]H.V. Seow
[1]Institute for Mathematical Research,
[2]Department of Mathematics, Faculty of Science,
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
[3]Nottingham University Business School, Faculty of Arts and Social Sciences,
University of Nottingham Malaysia Campus, Jalan Broga,
43500 Semenyih, Selangor, Malaysia

Abstract: Problem statement: The Capacitated Vehicle Routing Problem (CVRP) is a well-known combinatorial optimization problem which is concerned with the distribution of goods between the depot and customers. It is of economic importance to businesses as approximately 10-20% of the final cost of the goods is contributed by the transportation process. **Approach:** This problem was tackled using an Ant Colony Optimization (ACO) combined with heuristic approaches that act as the route improvement strategies. The proposed ACO utilized a pheromone evaporation procedure of standard ant algorithm in order to introduce an evaporation rate that depends on the solutions found by the artificial ants. **Results:** Computational experiments were conducted on benchmark data set and the results obtained from the proposed algorithms shown that the application of combination of two different heuristics in the ACO had the capability to improve the ants' solutions better than ACO embedded with only one heuristic. **Conclusion:** ACO with swap and 3-opt heuristic has the capability to tackle the CVRP with satisfactory solution quality and run time. It is a viable alternative for solving the CVRP.

Key words: Capacitated vehicle routing problem, ant colony optimizatio, heuristics, pheromone

INTRODUCTION

The Capacitated Vehicle Routing Problem (CVRP) concerns the design of a set of minimum cost routes, starting and ending at a single depot, for a fleet of vehicles to service a number of customers with known demands. Mathematically, it can be represented by a weighted graph G = (V, A) with V = {0,1, 2,…, n} as the vertex set and A = {(i, j) | i, j∈ V} as the edge set. The depot is denoted as vertex 0 and the total of n cities or customers to be served are represented by the other vertices. For each edge (i,j), i≠j, there is a nonnegative distance d_{ij} each measured using Euclidean computations. Each customer i, i=1,2,…,n, is associated with a nonnegative demand q_i and a service time δ_i which have to be satisfied. The demand at the depot is set to $q_0 = 0$ and its service time is set to $\delta_0 = 0$. Each vehicle is given a capacity constraint, Q. Consequently, the objective of the CVRP is to find a set of minimum cost routes to serve all the customers by satisfying the following constraints which are listed in Voss (1999): (i) each customer is visited exactly once by exactly one vehicle, (ii) all vehicle routes start and end at the depot, (iii) for each vehicle route, the total demand does not exceed the vehicle capacity Q and (iv) for each vehicle route, the total route length (including service times) does not exceed a given bound L.

Since the CVRP is a NP-hard problem, only instances of small sizes can be solved to optimality using exact solution methods (Toth and Vigo, 2002; Baldacci *et al*., 2010). As a result, heuristic methods are used to find good, but not necessarily guaranteed optimal solutions using reasonable amount of computing time. Starting with the simple constructive approaches such as the savings algorithm proposed by Clarke and Wright (1964) or basic improvement methods such as the 2-opt heuristic, the general-purpose heuristic methods (which are called metaheuristics) have then been developed to guide subordinate heuristics to avoid or overcome local optimality. During the past two decades, an increasing number of literatures on heuristic approaches have been developed to tackle the CVRP. The summary and discussion of several important and state-of-the-art modern heuristics for the problem can be found in the study by Cordeau *et al*. (2002) and Szeto *et al*. (2011).

The Ant Colony Optimization (ACO) was first introduced by Dorigo and Stutzle (2004). It is inspired

Corresponding Author: Lee, L.S., Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

by the real life behavior of ants foraging for food. During the search for food from their nest to the food source, it was found that a moving ant will lay a chemical substance called pheromone on the trail. The pheromone trail is a form of communication among the ants which will attract the other ants to use the same path to travel. Thus, higher amount of pheromone will enhance the probability of the next ant selecting that path to travel. With times, as more ants are able to complete the shorter path, the pheromone will accumulate faster on shorter path compared to the longer path. Consequently, majority of the ants would have travelled on the shortest path. Detailed descriptions of the ACO can be found in the book by Dorigo and Stutzle (2004). Recent applications of ACO can be found in Naganathan and Rajagopalan (2011) and Yap *et al*. (2012).

To apply the ACO for solving the CVRP, Voss (1999) first developed an ACO algorithm which is called Ant System (AS) for the problem and then presented an improved AS in Bullnheimer *et al.* (1999). Since then, many researchers have proposed new methods to improve the original ACO especially by applying other algorithms into the ACO to tackle the large-scaled CVRP. For instance, Doerner *et al.* (2002) proposed a hybrid approach for solving the CVRP by combining the AS with the savings algorithm. After that, Reimann *et al*. (2002) improved on the method in Doerner *et al*. (2002) by presenting a Savings based Ant System (SbAS) and then Reimann *et al.* (2004) proposed an approach called D-Ants which is competitive with the best Tabu Search (TS) algorithm in terms of solution quality and computation time. Also, Mazzeo and Loiseau (2004); Bell and McMullen (2004); Yu *et al.* (2009) and Zhang and Tang (2009), have made major contributions to the development of ACO to tackle the CVRP. This study aims to compare the solution quality of different basic heuristics combined with an original ACO in solving the problem.

MATERIALS AND METHODS

The main tasks considered in an ACO algorithm consist of the solution construction, the management of the pheromone trails and the additional techniques such as heuristic. Overall, the main procedures of the proposed ACO for solving the CVRP are summarized in a flowchart illustrated in Fig. 1.

Solution construction: In an ACO, each artificial ant simulates a vehicle and its complete set of routes is constructed by successively choosing customers to visit until all the customers have been visited. A new route will be started from the depot whenever the choice of the next customer to be visited leads to an infeasible solution due to the vehicle capacity or the total route length constraint. Consequently, there are a total of m solutions constructed sequentially by the total of m artificial ants in one run of iteration.

Initially, each ant is assigned to a randomly chosen customer as its first city to visit from the depot. Then, at each construction step, an ant k at current city i will select the next city j to visit from a feasible neighborhood N_i^k according to a probability distribution as in Eq. 1:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta [\mu_{ij}]^\gamma}{\sum_{l \in N_i^k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta [\mu_{ij}]^\gamma}, \text{ if } j \in N_i^k \quad (1)$$

where, $\eta_{ij} = 1/d_{ij}$ is a heuristic value, τ_{ij} denotes the pheromone concentration on the edge connecting cities i and j while $\mu_{ij} = d_{i0} + d_{0j} - d_{ij}$ is the savings of combining two cities i and j on one tour as opposed to visiting them on two different tours. The parameters α, β and γ bias the relative influence of the pheromone concentration, the heuristic value and the savings value. With the above probability in Eq. 1, the selection of a city that has not yet been visited would depend on the following criteria:

- Pheromone concentration, τ_{ij} which indicates how good the choice of the next city j from the current city i from the past
- Attractiveness, η_{ij} which indicates how promising the choice of the next city j is from current city i
- Savings, μ_{ij} which measures the favorability of combining two cities i and j to a tour where high savings indicate that visiting the next city j from current city i is a good choice
- Feasible neighborhood, N_i^k which is also called the candidate list where it includes only the closest cities for the current city i to be available for selection as the next city to be visited in the route

The probability of choosing a particular edge (i, j) will increase with the addition in the value of the corresponding pheromone concentration τ_{ij} whereas the values of the heuristic information η_{ij} and savings μ_{ij} will not dynamically change over time. However, when all the cities in the candidate list have already been visited by the ants, one city out of those not in the candidate list will be chosen.

In this case, an ant k will select the city (among the remaining cities) with maximum value of $[\tau_{ij}]^\alpha [\eta_{ij}]^\beta [\mu_{ij}]^\gamma$ as the next to move to. The use of the candidate list has the ability to significantly reduce the computation time necessary for the ants to construct solutions since the ants choose among a much smaller set of cities.

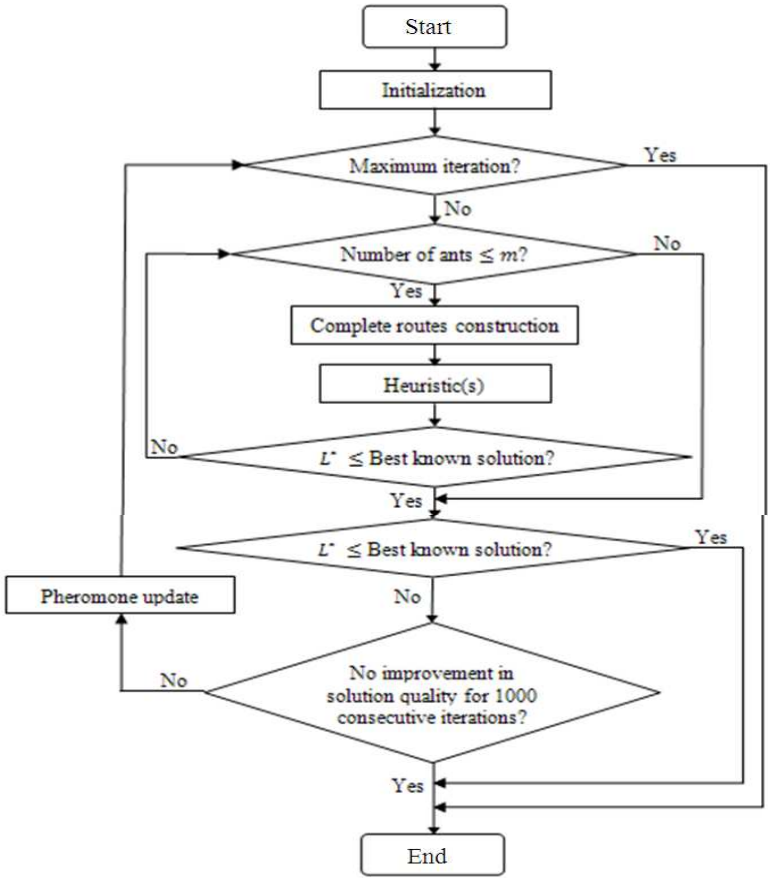

Fig. 1: Flowchart of ACO for the CVRP

But it should be noted that the use of a truncated candidate list can lead to not finding the optimal solution. During the routes construction process, an ant k returns to the depot when the carried quantity of demands meets the vehicle capacity constraint or the total route length constraint is violated. After that, the same ant k which represents a vehicle will start a new route again to serve the customers that have not yet been visited. This process will be repeated until all the customers have been visited.

Heuristics: After an artificial ant has finished constructing a solution but before the following ants start to build their solutions, the pheromone is updated and the ant's solution will be improved by applying a heuristic. There are four basic heuristics which are of interest to us:

Swap: This heuristic aims at improving the clustering of the solution by exchanging two customers from different routes, i.e., a customer i from route a is exchanged with a customer j from route b if there is an improvement of solution quality. In the proposed ACO, the swap heuristic will stop once there is a successful exchange of two cities between two different routes or there is no improvement found for the solution built.

Subtour reversal: This heuristic adjusts a sequence of cities to be visited in the current solution by selecting a subsequence of the cities and reversing the order. In detail, for an n-city situation, this heuristic starts with a feasible route and then tries to improve on it by reversing 2-city subtours, followed by 3-city subtours and continuing until reaching subtour of size n-1. The improvement is based on the largest decrease in travelled distance and the ties will be solved randomly. The stopping rule is subjected to when there is no subtour reversal improving the vehicle routes in the solution.

2-opt: This heuristic is applied separately to each of the vehicle routes built by an ant. Starting from a feasible route, it modifies the current route by deleting two edges, reversing one of the resulting paths and then reconnecting the route with two new edges. In the proposed ACO, the 2-opt is implemented to each vehicle route by using the best-improvement stopping rule.

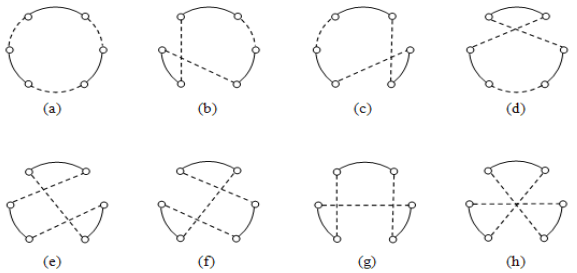

Fig. 2: All possible 3-opt recombination cases

3-opt: Three edges of a tour are removed in a 3-opt move and a new tour is obtained by replacing at most three of its arcs. In this context, the removal of three edges will result in three paths that can then be recombined into a full tour in eight different ways as shown in Fig. 2. However, only four (e, f, g, h) of the eight ways actually introduce three new edges while the other four ways (a, b, c, d) can be obtained by the 2-opt move.

Different combination of the above mentioned four heuristics would be applied to the original ACO during the computational experiments to determine the best combination of the heuristics with ACO in solving the CVRP.

Pheromone update: After all the artificial ants have improved the solutions through the heuristics, the pheromone trails will be updated. This is the main feature of an ACO algorithm which assists at improving future solutions since the updated pheromone trails would reflect the ants' performance and the quality of their solutions found. In this context, there are two main phases of the pheromone update in an AS algorithm (Dorigo and Stutzle, 2004), which are the pheromone evaporation and the pheromone deposition. In the proposed ACO, modifications would be made to the usual pheromone evaporation whereas the pheromone deposition would be referred to Bullnheimer *et al.* (1999) which comprises of the elitist strategy and also the concept of ranking. The details of the pheromone update procedures implemented in the proposed ACO are described as follows:

Pheromone evaporation: First of all, the pheromone concentration on all edges will be lowered by a constant factor with the following Eq. 2:

$$\tau_{ij} = \left(\rho + \frac{\theta}{L^{avg}}\right)\tau_{ij},\ \forall (i,j) \in A \qquad (2)$$

where, $0 \le \left(\rho + \frac{\theta}{L^{avg}}\right) < 1$ is an evaporation factor, $0 \le \rho < 1$ is the trail persistence, θ is a constant, $L^{avg} = \sum_{k=1}^{m} \frac{L^k}{m}$ is the average total distance found by m artificial ants per iteration and L^k is the total distance obtained by the solution of an artificial ant k. The ACO utilizes $\left(\rho + \frac{\theta}{L^{avg}}\right)$ as the evaporation factor as opposed to the pheromone evaporation in an AS algorithm from Dorigo and Stutzle (2004) which uses only the ρ as the trail persistence. The idea is to simulate the evaporation process of the pheromone trail in nature which depends on the length of the path travelled by an ant. The longer the path is, the more pheromone evaporates. Consequently, it favors the exploration of not yet visited edges by making the edges already visited by the ants less attractive. Furthermore, this process can avoid early or quick convergence of all the ants toward a suboptimal solution.

Pheromone deposition: After the pheromone evaporation process, only the best ants and the elitist ants will deposit pheromone on the edges that they have travelled following the Eq. 3 below:

$$\tau_{ij} = \tau_{ij} + \sum_{\lambda=1}^{\sigma-1} \Delta\tau_{ij}^{\lambda} + \Delta\tau_{ij}^{*}, \qquad (3)$$

Where:

$$\Delta\tau_{ij}^{\lambda} = \begin{cases} \frac{(\sigma-\lambda)}{L^{\lambda}}, & \text{if the } \lambda\text{th best ant travels on edge (i, j)} \\ 0, & \text{otherwise} \end{cases}$$

$$\Delta\tau_{ij}^{*} = \begin{cases} \frac{\sigma}{L^{*}}, & \text{if edge (i, j) is part of the best solution} \\ 0, & \text{otherwise} \end{cases}$$

Following Bullnheimer *et al.* (1999), two types of pheromone trails are laid during the pheromone update process with the Eq. 3. Firstly, the best-so-far solution (objective value L^*) found since the start of the ACO algorithm will be updated as if σ elitist ants had traversed it. The quantity of the pheromone deposited by the elitist ants is $\Delta\tau_{ij}^{*}$. Secondly, only the σ-1 best ants out of m ants of the current iteration are allowed to lay pheromone on the edges that they have traversed. The amounts of pheromone laid by these ants depend on their rank λ and also their solution quality L^{λ}, where the λth best ant lays an amount of pheromone equals to $\Delta\tau_{ij}^{\lambda}$. In short, the idea of the elitist strategy is to provide strong additional reinforcement to the edges belonging to the best solution found so far after every run of iteration. The aim is to guide the search in succeeding iterations as it is likely that some edges of the best-so-far solution are part of the optimal solution. On the other hand, the concept of ranking which is

suggested in Bullnheimer *et al.* (1997) aims to avoid the danger of over-emphasized pheromone trails caused by many ants using suboptimal routes.

RESULTS

The computational experiments were performed on a set of benchmark problems which are publicly available at the VRPWeb at: http://neo.lcc.uma.es/radi-aeb/WebVRP/.

Table 1: Characteristics of the benchmark problem instances

Problem	n	Q	L	δ	Vehicle used	Best published
Random problem						
C1	50	160	∞	0	5	524.61
C2	75	140	∞	0	10	835.26
C3	100	200	∞	0	8	826.14
C4	150	200	∞	0	12	1028.42
C5	199	200	∞	0	17	1291.29
C6	50	160	10	200	6	555.43
C7	75	140	10	160	11	909.68
C8	100	200	10	230	9	865.94
C9	150	200	10	200	14	1162.55
C10	199	200	10	200	18	1395.85
Clustered problem						
C11	120	200	∞	0	7	1042.11
C12	100	200	∞	0	10	819.56
C13	120	200	50	720	11	1541.14
C14	100	200	90	1040	11	866.37

n: number of customers δ: service time L: maximum tour length Q: vehicle capacity

These fourteen vehicle routing test problems have been widely used as benchmarks and their characteristics are summarized in Table 1.

From the initial investigation, we observed that the following parameter settings give a good compromise between the computation time and the solution quality for the proposed ACO:

- m = n artificial ants
- $\alpha = 2$, $\beta = 5$, $\gamma = 9$
- $\rho = 0.80$, $\theta = 80$
- candidate list size of $\lceil n/3 \rceil$
- $\sigma = 3$ elitist ants

Besides, with the suggestion from Dorigo and Stutzle (2004), the initial pheromone concentration was set as $\tau_0 = m/L^{nn}$, where L^{nn} is the total length of the solution generated by the nearest-neighbor heuristic. This is due to the fact that it is a good practice to set the initial pheromone concentration to a value that is slightly higher than the expected amount of pheromone deposited by the ants in one iteration. For all the problems tested, we set the maximum iteration to 50000.

Table 2: Computational results of original ACO and ACOs with one heuristic

				ACO with heuristic											
	Original ACO			Swap			Subtour Reversal			2-opt			3-opt		
Prob.	Best RPD	Avg RPD	Time (sec)	Best RPD	Avg. RPD	Time (sec)	Best RPD	Avg RPD	Time (sec)	Best RPD	Avg RPD	Time (sec)	Best RPD	Avg RPD	Time (sec)
C1	0.20	4.61	4.03	0.64	3.07	10.83	0.04	3.77	7.51	0.00	1.01	25.55	0.00	0.87	114.04
C2	4.09	6.32	8.15	4.79	6.33	18.40	3.31	4.99	9.78	2.69	3.53	7.83	1.44	2.68	38.91
C3	6.48	7.79	15.32	6.11	7.77	43.17	5.72	6.94	15.98	3.19	4.65	21.71	2.34	3.94	282.87
C4	10.04	12.30	38.01	10.43	11.88	107.97	8.74	10.63	53.70	7.44	9.00	49.09	6.17	7.63	546.61
C5	13.12	14.88	87.03	12.67	14.06	183.79	12.41	13.74	94.86	10.65	11.99	108.12	9.57	10.49	863.89
C6	0.85	3.48	5.92	1.22	3.24	9.66	1.34	3.36	7.46	1.39	2.91	7.65	0.77	2.22	24.43
C7	6.32	8.16	10.20	4.81	7.57	16.43	5.19	7.14	9.21	4.15	5.01	10.31	4.08	5.24	17.97
C8	7.76	10.09	16.16	6.36	8.36	31.02	5.16	7.32	22.38	4.52	6.15	18.54	1.29	3.63	139.46
C9	10.85	13.07	47.93	11.31	13.75	72.15	11.12	12.27	62.02	9.44	11.17	47.12	6.58	8.73	334.98
C10	12.76	15.90	122.63	14.59	16.24	182.06	13.39	14.65	119.96	11.68	12.99	104.36	10.50	11.56	688.65
C11	5.00	6.07	20.92	3.63	5.35	77.62	3.95	4.90	26.94	1.83	2.51	32.01	1.09	1.35	750.10
C12	6.22	8.43	17.47	7.16	8.60	28.38	6.23	7.24	18.15	3.23	5.06	15.69	3.44	3.88	204.36
C13	3.34	5.33	22.70	4.93	5.28	52.12	3.68	4.74	25.19	3.02	3.59	23.14	1.98	2.53	301.19
C14	1.42	2.45	15.89	1.41	2.54	38.30	1.60	1.97	16.94	0.43	0.64	14.43	0.28	0.36	116.27
Avg.	6.32	8.49	30.88	6.43	8.15	62.28	5.85	7.40	35.01	4.55	5.73	34.68	3.54	4.65	315.98

Table 3: Computational results of ACOs with two heuristics

	ACO with heurisitc								
	Swap and Subtour Reversal			Swap and 2-opt			Swap and 3-opt		
Prob.	Best RPD	Avg. RPD	Time (sec)	Best RPD	Avg. RPD	Time (sec)	Best RPD	Avg. RPD	Time (sec)
C1	0.04	2.49	18.96	0.00	0.76	40.45	0.00	0.48	412.74
C2	4.16	5.17	15.09	2.48	3.27	16.27	1.27	2.12	55.80
C3	4.88	6.46	38.35	3.36	4.84	40.73	1.47	3.43	285.53
C4	9.64	10.77	98.94	7.99	8.90	121.75	4.94	6.64	630.57
C5	11.81	13.27	205.50	9.86	11.23	177.26	8.17	9.71	1148.17
C6	1.16	2.84	8.03	1.37	2.50	19.49	0.00	1.32	41.34
C7	4.10	6.14	19.33	3.46	5.05	12.68	2.21	4.04	34.35
C8	4.52	7.18	43.92	1.98	4.07	42.17	1.07	2.94	163.13
C9	10.99	12.33	80.41	8.58	10.57	79.73	5.46	8.65	336.35
C10	13.24	15.12	173.49	11.51	12.58	167.33	9.25	11.25	747.34
C11	3.26	4.24	63.50	1.31	2.41	84.62	0.77	1.29	787.97
C12	5.69	7.12	31.93	4.54	5.35	34.24	2.05	3.55	242.51
C13	3.51	4.09	58.29	2.34	3.14	50.03	1.90	2.51	269.98
C14	1.38	1.81	39.28	0.26	0.62	30.67	0.15	0.30	143.68
Avg.	5.60	7.07	63.93	4.22	5.38	65.53	2.77	4.16	378.53

The computational experiments were conducted on a desktop computer of Intel (R) Core2 Duo processor with 3GB RAM and 3GHz CPU speed. The algorithms were coded in ANSI-C using Microsoft Visual C++ Version 6 as the compiler. Each algorithm was run 10 times on each instance tested. The results which are reported for each instance are computed according to the Eq. 4 below:

$$RPD = \frac{Method_{sol} - Best_{sol}}{Best_{sol}} \times 100\% \quad (4)$$

where, RPD is the relative percentage deviation of the solution obtained (denoted as $Method_{sol}$) from the best published solution (denoted as $Best_{sol}$).

The computational results are presented in Table 2 and Table 3. For each instance in both Table 2 and 3, the results are presented in the form of RPD (both for best solution obtained and on average) and the average run time is shown in CPU seconds. A RPD value of 0.00% indicates that the best published solution of the total distance is obtained. The last row of both of the tables shows the average RPD and average run time over all instances tested for different approaches.

DISCUSSION

Table 2 shows the results obtained by an original ACO algorithm and those ACO algorithms combined with one heuristic. In addition, we further extended the investigation on ACOs by combining two heuristics into the original ACO algorithm with the results in Table 3. There are a total of three combinations of two different heuristics considered in our study since only the swap algorithm functions at improving the clustering with the modification among different vehicle routes whereas the other three heuristics function at improving the routing with the alteration within one vehicle route.

All the ACOs with heuristic perform better than the original ACO with respect to the solution quality. However, ACOs with heuristic consume more run time compared to the original ACO especially for the ACOs involving the application of 3-opt heuristic. Nevertheless, the ACO with 3-opt heuristic performs the best in Table 2 with an average RPD of 4.65% and the ACO with swap and 3-opt heuristics also outperform other ACOs in Table 3 with an average RPD of 4.16%. From Table 2, it could be observed that the performance of the ACOs could be summarized as the application of 3-opt performs the best, followed by 2-opt, subtour reversal and then the swap heuristic. The same case for the results shown in Table 3 where the ACO with swap and 3-opt performs better than the ACO with swap and 2-opt following by the ACO with swap and subtour reversal. Besides, it is shown in Table 2 and Table 3 that the application of two different heuristics in the ACO has the capability to improve the ants' solutions better than the ACO embedded with only one heuristic. For instance, the ACO with swap and 2-opt obtains an average RPD of 5.38% whereas ACO with swap and ACO with 2-opt only reach the average RPD of 8.15% and 5.73% respectively.

CONCLUSION

The CVRP has been an attractive issue in the field of distribution and logistics which is motivated by both its practical relevance and its considerable difficulty. In this study, we have compared the solution quality of different basic heuristics combined with an original ACO in solving the problem. The computational results of fourteen benchmark problems shown that the ACO combined with the swap and 3-opt heuristic has the capability to tackle the CVRP with satisfactory solution quality and run time. Therefore it is a viable alternative for solving the CVRP.

ACKNOWLEDGEMENT

This research is supported by Fundamental Research Grant Scheme (FRGS) 01-04-10-886FR (Ministry of Higher Education, Malaysia) The authors are also grateful to Universiti Putra Malaysia for providing the excellent research facilities.

REFERENCES

Baldacci, R., P. Toth and D. Vigo, 2010. Exact algorithms for routing problems under vehicle capacity constraints. Ann. Oper. Res., 175: 213-245. DOI: 10.1007/s10479-009-0650-0

Bell, J.E. and P.R. McMullen, 2004. Ant colony optimization techniques for the vehicle routing problem. Adv. Eng. Inform., 18: 41-48. DOI: 10.1016/j.aei.2004.07.001

Bullnheimer, B., R.F. Hartl and C. Strauss, 1997. A new rank based version of the Ant System. A computational study. University of Economics and Business, Vienna.

Bullnheimer, B., R.F. Hartl and C. Strauss, 1999. An improved Ant System algorithm for theVehicle Routing Problem. Ann. Oper. Res., 89: 319-328. DOI: 10.1023/A:1018940026670

Clarke, G. and J.W. Wright, 1964. Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res., 12: 568-581.

Cordeau, J.F., M. Gendreau, G. Laporte, J.Y. Potvin and F. Semet, 2002. A guide to vehicle routing heuristics. J. Oper. Res. Soc., 53: 512-522. DOI: 10.1057/palgrave/jors/2601319

Doerner, K., M. Gronalt, R.F. Hartl, M. Reimann and C. Strauss *et al*., 2002. SavingsAnts for the vehicle routing problem. Lecture Notes Comput. Sci., 2279: 73-109. DOI: 10.1007/3-540-46004-7_2

Dorigo, M. and T. Stutzle, 2004. Ant Colony Optimization. 1st Edn., MIT Press, Cambridge, ISBN-10: 0262042193, pp: 305.

Mazzeo, S. and I. Loiseau, 2004. An ant colony algorithm for the capacitated vehicle routing. Elect. Notes Dis. Math., 18: 181-186. DOI: 10.1016/j.endm.2004.06.029

Naganathan, E.R. and S. Rajagopalan, 2011. Hybrid traffic management model for multi-protocol label switching network. Am. J. Applied Sci., 8: 1322-1327. DOI: 10.3844/ajassp.2011.1322.1327

Reimann, M., K. Doerner and R.F. Hartl, 2004. D-ants: Savings based ants divide and conquer the vehicle routing problem. Comput. Oper. Res., 31: 563-591. DOI: 10.1016/S0305-0548(03)00014-5

Reimann, M., M. Stummer and K. Doerner, 2002. A savings based ant system for the vehicle routing problem. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO' 02), ACM, San Francisco, pp: 1317-1325.

Szeto, W.Y., Y. Wu and S.C. Ho, 2011. An artificial bee colony algorithm for the capacitated vehicle routing problem. Eur. J. Oper. Res., 215: 126-135. DOI: 10.1016/j.ejor.2011.06.006

Toth, P. and D. Vigo, 2002. The Vehicle Routing Problem. 1st Edn., Society for Industrial and Applied Mathematics, Philadelphia, ISBN-10: 0898715792, pp: 385.

Voss, S., 1999. Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. 1st Edn., Kluwer, Boston, ISBN-10: 0792383699, pp: 511.

Yap, C.N., L.S. Lee, Z.A. Majid and H.V. Seow, 2012. Ant colony optimization for container loading problem. J. Math. Stat., 7: 165-171. DOI: 10.3844/jmssp.2012.165.171

Yu, B., Z.Z. Yang and B.Z. Yao, 2009. An improved ant colony optimization for vehicle routing problem. Eur. J. Oper. Res., 196: 171-176. DOI: 10.1016/j.ejor.2008.02.028

Zhang, X. and L. Tang, 2009. A new hybrid ant colony optimization algorithm for the vehicle routing problem. Patt. Rec. Let., 30: 848-855. DOI: 10.1016/j.patrec.2008.06.001