

## Ant Colony Optimization for Solving Solid Waste Collection Scheduling Problems

Z. Ismail and S.L. Loh  
Department of Mathematics, Faculty of Science,  
University Technology Malaysia, 81310 Skudai, Johor, Malaysia

---

**Abstract: Problem statement:** Southern Waste Management environment (SWM environment) is a company responsible for the collection and disposal of solid waste for the city of Johor Bahru, a city with over one million populations. The company is implementing an integrated solid waste management system where it involved in the optimization of resources to ensure the effectiveness of its services. Formulating this real life problem into vehicle routing problem with stochastic demand model and using some designed algorithms to minimize operation cost of solid waste management. **Approach:** The implementation of Ant Colony Optimization (ACO) for solving solid waste collection problem as a VRPSD model was described. A set of data modified from the well known 50 customers problems were used to find the route such that the expected traveling cost was minimized. The total cost was minimized by adopting a preventive restocking policy which was trading off the extra cost of returning to depot after a stock-out with the cost of returning depot for restocking before a stock-out actually occurs. For comparison purposes, Simulated Annealing (SA) was used to generate the solution under the same condition. **Results:** For the problem size with 12 customers with vehicle capacity 10 units, both algorithms obtained the same best cost which is 69.4358 units. But the percentage deviations of averages from the associated best cost are 0.1322 and 0.7064 for ACS and SA. The results indicated that for all demand ranges, proposed ACO algorithm showed better performance than SA algorithm. **Conclusion:** SA was able to obtain good solutions for small ranges especially small size of problem. For ACS, it is always provide good results for all tested ranges and problems sizes of the tested problem.

**Key words:** Ant colony optimization, VRP, VRPSD, solid waste collection problems

---

### INTRODUCTION

Southern Waste Management Sdn. Bhd. (SWM) was established in line with Malaysian Government's decision on the National Privatization of Solid Waste Management. It was subsequently awarded the task of managing the storage, collection, transfer, haul, intermediate processing and disposal of solid waste in the Southern Region of Peninsular Malaysia by Economic Planning Unit of the Prime Minister's Department on Dec 21st, 1995. SWM is structured to provide the Southern Region with an integrated waste management system that is comprehensive, well planned, efficient and cost effective. Applying the most environmentally sound technology available, SWM will strive to enhance the quality of solid waste management and cleaning services in the country.

In April 1996, SWM was directed by the Government of Malaysia to take over the solid waste Management and public cleaning services from all

Local Authorities within the SWM concession area on an Interim basis. It is a transition towards the implementation of the full privatization. The concession area covers the states of Negeri Sembilan, Melaka and Johor. The concession will cover a period of 20 year and serve a population of 4 million people.

To improve their services, a strong KPI (Key Performance Indicator) will be instituted against the two consortiums to ensure they deliver high quality performance to the public. The objective is to provide a more efficient management of solid wastes and public cleaning to the public's satisfaction.

In this study, integrated solid waste management system is emphasized. SWM is involved in the integrated solid waste management system and public cleaning services. They will implement a strategic and well-rounded integrated system to ensure the effectiveness of its services. They strive to provide

---

**Corresponding Author:** Zuhaimy Ismail, Department of Mathematics, Faculty of Science, University Technology Malaysia, 81310 UTM Skudai, Johor, Malaysia Tel: +60197133940 Fax: +6075566162

optional solution for integrated waste management by drawing on the experience and expertise of their staff at various officers. They specialize in:

- Handling household waste, commercial and non-toxic industrial waste:
  - Waste containment to collection
  - Transfer station to landfills
  - Recycling and waste to energy

**Problem:** The solid waste collection problem is formulated into VRPSD model where a street or some nearby streets which had been group together as one unit is represented by the nodes (customers' locations). The customers' locations are in form of Cartesian coordinate where each point appears uniquely in a plane through two numbers, called x-coordinate and y-coordinate while demands of customers which are stochastic are recorded in a range form.

This study shows the way of a real life problem being formulated into Vehicle Routing Problem with Stochastic Demand Model and solved by the designed algorithm. Since there is no open source data for VRPSD problem, thus this case study adopted a set of data modified from the well known 50-customer problem in Eilon *et al.*<sup>[7]</sup> as in Table 1.

The customers' locations are shown on a plane by designed program as in Fig. 1 to show the positions of all customers graphically.

The problem described above can be modeled as a Vehicle Routing Problem with Stochastic Demand (VRPSD) as the amount of solid waste is stochastic and may be presented in a complete graph<sup>[7]</sup> and Secomandi<sup>[9,10]</sup>. Let the set of nodes be  $\{0, 1, \dots, n\}$ .

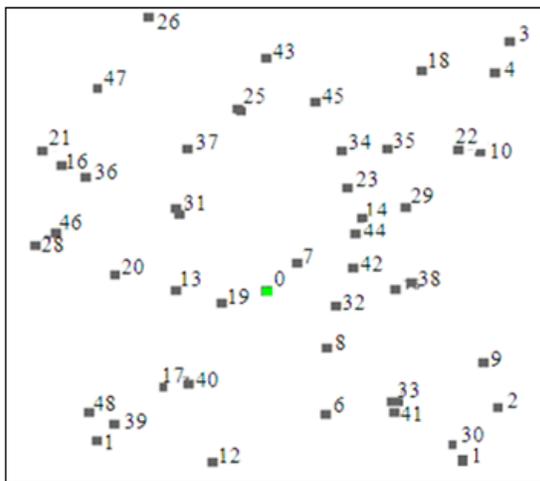


Fig. 1: Positions of customers' locations

Node 0 denoting the depot node and  $V = \{1, \dots, n\}$  is a set of customer locations. Distances between nodes are assumed to be symmetric and it satisfies the triangle inequality. Customers' demands are stochastic variables,  $\xi_i = 1, 2, \dots, n$  independently distributed with known distribution. It follows a discrete probability distribution with  $v$  possible values,  $\xi^1, \xi^2, \dots, \xi^v$  denoted by  $p_{i,k} = \text{Prob}(\xi_i = \xi^k)$ . Actual demand is only known when the vehicle arrives at the customer location.

Table 1: Data of case study

I	Customers' locations		$\xi_i$
	x-coor.	y-coor.	
0	5.00	4.00	-
1	1.89	0.77	4-7
2	9.27	1.49	2-9
3	9.46	9.36	3-5
4	9.20	8.69	0-4
5	7.43	1.61	5-7
6	6.08	1.34	0-8
7	5.57	4.60	3-5
8	6.10	2.77	3-6
9	8.99	2.45	1-7
10	8.93	7.00	6-7
11	8.60	0.53	5-7
12	4.01	0.31	1-6
13	3.34	4.01	3-7
14	6.75	5.57	4-7
15	7.36	4.03	1-2
16	1.24	6.69	1-3
17	7.13	1.92	1-7
18	7.86	8.74	4-6
19	4.18	3.74	5-9
20	2.22	4.35	5-6
21	0.88	7.02	5-8
22	8.53	7.04	1-5
23	6.49	6.22	1-8
24	4.53	7.87	1-3
25	4.46	7.91	0-4
26	2.83	9.88	3-5
27	3.39	5.65	1-6
28	0.75	4.98	0-5
29	7.55	5.79	1-4
30	8.45	0.69	1-6
31	3.33	5.78	3-7
32	6.27	3.66	3-8
33	7.31	1.61	0-6
34	6.37	7.02	1-6
35	7.23	7.05	0-7
36	1.68	6.45	1-4
37	3.54	7.06	2-8
38	7.67	4.17	0-6
39	2.20	1.12	3-7
40	3.57	1.99	2-8
41	7.34	1.38	0-4
42	6.58	4.49	1-6
43	5.00	9.00	1-4
44	6.63	5.23	3-4
45	5.89	8.06	1-8
46	1.13	5.25	2-6
47	1.90	8.35	3-7
48	1.74	1.37	2-9

After the vehicle completed service at customer  $w$ , suppose the vehicle has remaining load  $q$  and let  $f_w(q)$  denotes the expected cost from node  $w$  onward until the  $n$  customer and the way back to depot. With this notation, the expected cost of the a priori tour is  $f_0(Q)$ . The dynamic programming recursion is shown as follows<sup>[11]</sup>:

$$f_w(q) = \min \{f_w^p(q), f_w^r(q)\} \quad (1)$$

Where:

$$f_w^p(q) = c_{w,w+1} + \sum_{k:\xi^k \leq q} f_{w+1}(q - \xi^k) p_{w+1,k} + \sum_{k:\xi^k > q} [b + 2c_{w+1,0} + f_{w+1}(q + Q - \xi^k)] p_{w+1,k} \quad (2)$$

and

$$f_w^r(q) = c_{w,0} + c_{0,w+1} + \sum_{k=1}^v f_{w+1}(Q - \xi^k) p_{w+1,k} \quad (3)$$

with the boundary condition:

$$f_n(q) = c_{n,0}, \quad q \in S_n \quad (4)$$

Equation 2 and 3 represents the cost of proceeding directly to the next customer and the cost of the restocking action respectively.

Assume that one single vehicle with fixed capacity  $Q$  departs from the depot node to delivery goods at different customer locations according to their demands and at the same time, it has to minimize the total expected distance traveled. After all the demands have been served, the vehicle returns to the depot.

The vehicle visits the customers according to the sequence in the given a priori tour. It has to choose depending on the customer's demand either proceed directly to the next customer or return to depot for restocking. Thus, the goal of this study is find a vehicle route and a routing policy (threshold) at each node in order to minimize the total expected cost<sup>[1]</sup>.

## MATERIALS AND METHODS

**The ant colony system algorithm for solving VRPSD:** Ant System was efficient in discovering good or optimal solutions for small problems with nodes up to 30. But for larger problems, it requires unreasonable time to find such a good result. Thus, Dorigo and Gambardella<sup>[3,4]</sup> and Bianchi *et al.*<sup>[1]</sup> devised three main changes in Ant System to improve its performance which led to the existence of ant colony system.

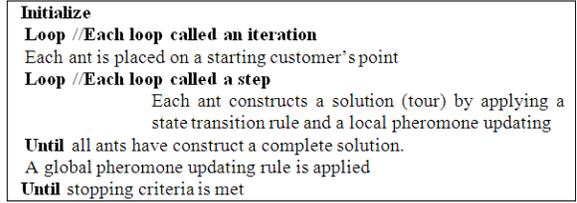


Fig. 2: Ant colony system's algorithm

Ant Colony System is different from Ant System in three main aspects. Firstly, state transition rule gives a direct way to balance between exploration of new edges and exploitation of a priori and accumulated information about the problem. Secondly, global updating rule is applied only to those edges which belong to the best ant tour and lastly, while ants construct the tour, a local pheromone updating rule is applied.

Basically, Ant Colony System (ACS) works as follows:  $m$  ants are initially positioned on  $n$  nodes chosen according to some initialization rule such as choosing by randomly, with at most one ant in each customer point. Each ant builds a tour incrementally by applying a state transition rule. While constructing the solution, ants also updating the pheromone on the visited edges by local updating rule. Once the ants complete their tour, the pheromone on edges will be updated again by applying global updating rule<sup>[2]</sup>.

During construction of tours, ants are guided by both heuristic information and pheromone information. Heuristic information refers to the distances of the edges where ants prefer short edges. An edge with higher amount of pheromone is a desirable choice for ants. The pheromone updating rule is designed so that ants tend to leave more pheromone on the edges which should be visited by ants.

The Ant Colony System algorithm is given as in Fig. 2<sup>[3,4]</sup>.

There are 3 main components which led to the definition of Ant colony System; they are state transition rule, global updating rule and local updating rule. Each of these components will be shown in detail as follow:

In the ant ACS, an artificial ant  $k$  after serves customer  $r$  chooses the customer  $s$  to move to from set of  $J_k(r)$  that remain to be served by ant  $k$  by applying the following state-transition rule which is also known as pseudo-random-proportional-rule:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \left\{ \begin{matrix} [\tau(r,u)] \\ [\eta(r,u)]^\beta \end{matrix} \right\} & \text{if } a \leq a_0 \quad (\text{exploitation}) \\ S & \text{otherwise (biased exploration)} \end{cases} \quad (5)$$

Where:

- $\beta$  = The control parameter of the relative importance of the visibility
- $\tau(r,u)$  = The pheromone trail on edge (r, u)
- $\eta(r,u)$  = A heuristic function which was chosen to be the inverse distance between customers r and u
- A = A random number uniformly distributed in [0,1]
- $a_0 (0 \leq a_0 \leq 1)$  = A parameter
- S = A random variable selected according to the probability distribution which favors edges which is shorter and higher amount of pheromone

It is same as in Ant system and also known as random-proportional-rule given as follow:

$$p_k(r,s) = \begin{cases} \frac{[\tau(r,s)].[\eta(r,s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r,u)].[\eta(r,u)]^\beta} & \text{if } s \in J_k(r) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where,  $p_k(r,s)$  is the probability of ant k after serves customer r chooses customer s to move to.

The parameter of  $a_0$  determines the relative importance of exploitation versus exploration. When an ant after serves customer r has to choose the customer s to move to, a random number  $a (0 \leq a \leq 1)$  is generated, if  $a \leq a_0$ , the best edge according to Eq. 5 is chosen, otherwise an edge is chosen according to Eq. 6<sup>[3]</sup>. While building a tour, ants visits edges and change their pheromone level by applying local updating rule as follow:

$$\tau(r,s) = (1 - \rho) \cdot \tau(r,s) + \rho \cdot \Delta\tau(r,s), \quad 0 < \rho < 1 \quad (7)$$

where:

- $\rho$  = A pheromone decay parameter
- $\Delta\tau(r,s) = \tau_0$  (initial pheromone level)

Local updating makes the desirability of edges change dramatically since every time an ant uses an edge will makes its pheromone diminish and makes the edge becomes less desirable due to the loss of some of the pheromone. In other word, local updating drives the ants search not only in a neighborhood of the best previous tour.

Global updating is performed after all the ants have completed their tours. Among the tours, only the best ant which produced the best tour is allowed to deposit

pheromone. This choice is intended to make the search more directed. The pheromone level is updated by global updating rule as follow:

$$\tau(r,s) = (1 - \alpha) \cdot \tau(r,s) + \alpha \cdot \Delta\tau(r,s), \quad 0 < \alpha < 1 \quad (8)$$

Where:

$$\Delta\tau(r,s) = \begin{cases} 1/L_{gb} & \text{if } (r,s) \in \text{global - best - tour} \\ 0 & \text{otherwise} \end{cases}$$

$L_{gb}$  is the length of the global best tour from the beginning of the trial. (global-best) and  $a$  is pheromone decay parameter. Global updating is intended to provide greater amount of pheromone to shorter tours. Eq. 8 dictates that only those edges belong to globally best tour will receive reinforcement.

From the experiments done by previous researchers, the numerical parameters are set as following values:

$$\beta = 2, a_0 = 0.9, \alpha = \rho = 0.1, \tau_0 = (n \cdot L_{nn})^{-1}$$

Where:

- $L_{nn}$  = The tour length produced by the nearest neighbor heuristic
- $n$  = The number of customers

The number of ants used is  $m = 10$ . These values were obtained by a preliminary optimization phase in which it is found that the experimental optimal values of the parameters were largely independent of the problem except for  $\tau_0$ <sup>[4]</sup>.

**Case Study:** A case study has been carried out and it shows the way of a real life problem being formulated into Vehicle Routing Problem with Stochastic Demand Model and solved by the designed algorithm. Since there is no open source data for VRPSD problem, thus this case study adopted a set of data modified from the well known 50-customer problem in Eilon *et al.*<sup>[6]</sup> as in Table 1.

For this case study, the stopping criterion for ACS algorithm is in dynamic form where either number of consecutive non-improving solutions reaching 50 or number of iteration reaches the maximum level. For SA algorithm, the stopping criterion used is in static form where the iteration stops after a predetermined number of maximum iteration is reached. The move that is used in SA algorithm to obtain the neighboring solution is the swapping of position of any two randomly chosen customers' positions. Parameter settings of ACS and SA algorithm for this case study are shown in Table 2 and 3 respectively.

Table 2: Parameter setting of ACS algorithm

Number of ants	10
Capacity of vehicle	10-50
$\alpha$	0.9
$\alpha, \rho$	0.1
$\beta$	2
$\tau_0$	$(n.L_{mn})^{-1}$
Maximum iterations	100
No. of consecutive non-improving solutions allowed	50

Table 3: Parameter setting of SA algorithm

Initial temperature	1000.00
Number of repetition	20.00
Cooling rate	0.98
Maximum iteration	1500.00

For this study, the proposed ACS with local search algorithm and Simulated Annealing (SA) algorithm are tested for problems ranging in size from 12-48 customers. Further detail on SA may be found in Kirkpatrick *et al.*<sup>[8]</sup>. The program is run for 10 trials for each algorithm under different vehicle capacity values from ten to forty. Best cost refers to the best cost found from the 10 trials for comparison purpose.

**RESULTS**

Table 4 shows the computational results for algorithm ACS and SA under different problem sizes.

The results indicate that proposed ACS with local search algorithm produces better solutions compared to SA algorithm. For smaller size of problem, deviation between the best cost results from both ACS and SA algorithm is relatively small. For the problem size with 12 customers with vehicle capacity 10 units, both algorithms obtained the same best cost which is 69.4358 units. But the percentage deviations of averages from the associated best cost are 0.1322 and 0.7064 for ACS and SA. This indicates that ACS gives the results in a more consistent manner compared to SA.

The deviation for SA's solution compared to solutions obtained by ACS is increasing with the problem sizes. For problem size with 12 customers, the best solutions of SA deviated from best solutions of ACS for not more than 0.5%. When it comes to the problem sizes with 48 customers, the best solutions of SA deviated from best solutions of ACS for more than 5%. This indicates that ACS performs much better than SA algorithm in bigger size of problem. However, ACS requires more computational effort or time over SA algorithm.

For each of the problem size, the larger of vehicle capacity gives the lower total expected cost. The reason behind is, the larger capacity of vehicle is able to satisfy more customers' demands in which it reduces the occurring times for preventive restocking and route failure.

Table 4: Comparison of algorithm ACS and SA for different problem sizes

Problem sizes, n	Capacity, Q	Total expected cost of a priori				Difference (Best ACS- best SA)
		ACS		SA		
		Best	Average	Best	Average	
12 (Model 1)	10	69.4358	69.5276	69.4358	69.9263	0.0000
	20	42.7758	42.7825	42.8547	43.2857	-0.0789
	30	37.2618	37.3537	37.3588	37.6974	-0.0970
	40	33.8966	33.9691	34.0416	34.7209	-0.1450
24 (Model 2)	10	114.7422	114.8451	116.6947	117.9591	-1.9525
	20	70.1857	70.5992	70.7853	72.9933	-0.5996
	30	55.0705	55.5264	55.4437	58.4765	-0.3732
	40	50.4478	50.9085	51.0495	53.1760	-0.6017
36 (Model 3)	10	154.1018	154.8055	156.9437	159.3486	-2.8419
	20	94.1029	94.9249	97.4506	99.4608	-3.3477
	30	73.5592	74.2729	77.1142	79.7676	-3.5550
	40	62.6947	63.2958	68.7049	71.6983	-6.0102
48 (Model 4)	10	199.2149	200.2451	209.8118	212.2541	-10.5969
	20	118.7979	119.5347	126.0159	127.6119	-7.2180
	30	91.1003	91.6274	96.3949	100.2883	-5.2946
	40	76.9768	78.0332	84.0350	87.9365	-7.0582

Table 5: Comparison of algorithm ACS and SA for different demand ranges

Problem sizes, n	Demand ranges (max-min)	Total expected cost of a priori				Difference (Best ACS- best SA)
		ACS		SA		
		Best	Average	Best	Average	
12 (Model 1)	5	37.8309	37.8309	37.8309	37.8946	0.0000
	10	46.5176	46.5233	46.5176	46.6858	0.0000
	15	55.9193	55.9193	55.9193	56.4056	0.0000
	20	66.3211	66.3906	66.4080	66.7586	-0.0869
24 (Model 2)	5	60.6567	60.8200	62.1815	63.3988	-1.5248
	10	79.9053	80.0058	81.5747	80.6062	-1.6694
	15	97.5548	97.6098	99.4316	101.3428	-1.8768
	20	116.1023	116.1148	117.3202	119.0287	-1.2179
36 (Model 3)	5	80.1849	80.8623	84.1031	85.5617	-3.9182
	10	106.9875	107.3627	110.6752	112.5322	-3.6877
	15	133.3562	133.5786	137.9571	141.2781	-4.6009
	20	160.7067	161.0139	161.1994	166.7799	-0.4927
48 (Model 4)	5	97.9829	98.7263	103.2490	105.2380	-5.2661
	10	133.5311	134.0055	140.2354	143.1984	-6.7043
	15	168.1742	168.8197	175.8930	179.7441	-7.7188
	20	205.0472	205.5534	214.8841	218.3807	-9.8369

In short, ACS showed better performance than SA under all of the problem sizes with various vehicle capacities tested. Next, both algorithms will be tested over demand ranges.

To access the robustness of ACS algorithm, both ACS and SA algorithms are further compared for different demand ranges. The customers' demands are generated ranged from 5-20. Again, the program is run for 10 trials for each algorithm under different demand ranges and problem sizes. The capacity of vehicle is set to be 30 units for every trial for this comparison. Table 5 shows the computational results for comparison.

The results indicate that for all demand ranges, proposed ACS algorithm showed better performance than SA algorithm. From Table 5, it can be noted that ACS gives the more consistent solutions compared to

SA algorithm. The average total expected cost given by ACS does not deviate too far from its best cost.

With a fixed capacity, the deviation of best costs for SA algorithm from the best cost of ACS is increasing as the demand ranges increases. Furthermore, for problem size with 12 customers; the best solutions for SA deviated from best solutions of ACS for not more than 0.2%. However, for problem size with 48 customers, the best solutions of SA deviated from best solutions of ACS for more than 4.5%. This shows that ACS algorithm can reach a good solution for larger problem size compare to SA algorithm.

Apparently, SA is able to obtain good solutions for small ranges especially small size of problem. For ACS, it is always provide good results for all tested ranges and problems sizes of the tested problem.

### DISCUSSION

Generally, Genetic Algorithm gives a pool of solutions rather than just one. The process of finding superior solutions mimics the evolution process, with solutions being combined or mutated to find out the pool of solutions. Simulated Annealing is a global optimization technique which traverses the search space by generating neighboring solutions of the current solution. A superior neighbor is always accepted and an inferior neighbor is accepted with some probability.

Tabu Search is similar to Simulated Annealing, in that both traverse the solution space by testing mutations of an individual solution. However, simulated annealing generates only one mutated solution but Tabu search generates many mutated solutions and moves to the solution with the lowest fitness of those generated.

Ant Colony System is the extension from Ant System. Both algorithms are categorized as Ant Colony Optimization (ACO) algorithms. In particular, it can be observed that ACS is the most aggressive of the ACO algorithms and it returns the best solution quality for very short computation times<sup>[5,7]</sup>. ACS has an advantage over Simulated Annealing and Genetic Algorithm approaches when the graph may change dynamically where the Ant Colony algorithm can be run continuously and adapt to changes in real time.

From the results of case study above, it can be known that the ACS algorithm always finds very good solutions for all tested problems in the aspects of various problem sizes and demand ranges. The algorithm finds the good solutions efficiently and consistently compare with other heuristic methods such as Simulated Annealing and it does not exhibit stagnation behavior where the ants continue to search

for new possibly better tours. Stagnation behavior is the situation in which all ants make the same tour.

### CONCLUSION

The proposed ACS with local search algorithm and Simulated Annealing algorithm are tested for problems ranging in size from 12-48 customers. The results indicate that proposed ACS with local search algorithm produces better solutions compare to SA algorithm. Deviation of SA algorithm's best cost from the best cost results of ACS algorithm is increasing as the problem size increases.

In the second part of comparison, both ACS and SA algorithms are further compared for different demand ranges from 5-20 units. Again, the results indicate that proposed ACS algorithm showed better performance than SA algorithm. With a fixed capacity, the deviation of best costs for SA algorithm from the best cost of ACS is increasing as the demand ranges increases.

In this study, an efficient heuristic method is presented to solve the VRPSD problem. The ACS exploits the nature phenomenon of ants to solve such stochastic optimization problem. A local search algorithm (descent method) is proposed to be added in the ACS algorithm. The concept of this method is to find the a priori tour that gives the minimum total expected cost. The ACS has been shown the ability in obtaining good solution for VRPSD problem.

### ACKNOWLEDGEMENT

This research was supported by the Ministry of Science, Technology and Innovation, Malaysia (MOSTI) under IRPA Grant Vot No. 74285 and the Department of Mathematics, Faculty of Science, University Technology Malaysia. These supports are gratefully acknowledged. The authors would like to thank lecturers and friends for many helpful ideas and discussion.

### REFERENCES

1. Bianchi, L., M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli and T. Schiavinotto *et al.*, 2006. Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Math. Model. Algorithms*, 5: 91-110. <http://www.citeulike.org/group/1786/article/904938>
2. Dorigo, M. and G.D. Caro, 1999. The Ant Colony Optimization Meta-Heuristic. In: *New Ideas in Optimization*, Corne, D., M. Dorigo and F. Glover (Eds.), McGraw-Hill, London, ISBN: 0-07-709506-5, pp: 11-32.

3. Dorigo, M. and L.M. Gambardella, 1997. Ant colonies for the traveling salesman problem. *Biosystem*, 43:73-81.  
<http://www.ncbi.nlm.nih.gov/pubmed/9231906>
4. Dorigo, M. and L.M. Gambardella, 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evolut. Comput.*, 1: 53-66. DOI: 10.1109/4235.585892
5. Dorigo, M. and T. Stutzle, 2004. *Ant Colony Optimization*. MIT Press, Cambridge, ISBN: 0262042193, pp: 305.
6. Eilon, S., C.D.T. Watson-Gandy and N. Christofides, 1971. *Distribution Management: Mathematical Modeling and Practical Analysis*. Compton Printing Ltd., Great Britain. ISBN: 9780852641910.
7. Ismail, Z. and Irhamah, 2008. Solving the vehicle routing problem with stochastic demands via hybrid genetic algorithm-tabu search. *J. Math. Stat.*, 4: 161-167.  
<http://www.scipub.org/fulltext/jms2/jms243161-167.pdf>
8. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, 1983. Optimization by simulated annealing. *Science*, 220: 671-680.  
<http://www.ncbi.nlm.nih.gov/pubmed/17813860>
9. Secomandi, N., 2000. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Comput. Operat. Res.*, 27: 1171-1200.  
<http://portal.acm.org/citation.cfm?id=350215>
10. Secomandi, N., 2001. A rollout policy for the vehicle routing problem with stochastic demands. *Operat. Res.*, 49: 796-802.  
<http://portal.acm.org/citation.cfm?id=769023>
11. Yang, W.H., M. Kamlesh and H.B. Ronald, 2000. Stochastic vehicle routing problem with restocking. *Transport. Sci.*, 34: 99-112.  
<http://portal.acm.org/citation.cfm?id=767755.768835>