Suitability Factor on the Capacitated Vehicle Routing Problem

Helmi Md Rais
Computer and Information Science Department
Universiti Teknologi PETRONAS
Malaysia
helmim@petronas.com.my

Zulaiha Ali Othman and Abdul Razak Hamdan
Centre for Artificial Intelligence Technology
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
Malaysia

zao@ftsm.ukm.my; arh@ftsm.ukm.my

Abstract - **Finding a good solution for logistics and transportation industries is a continuous effort to maximize the efficiencies especially on problem that relates to the industry such as Vehicle Routing Problem (VRP). One of the most popular solutions is using Ant Colony System (ACS) algorithm. Several versions of ACOs have been proposed which aim to achieve an optimum solution. A new variant of algorithm called Dynamic Ant Colony System with Three Level Updates (DACS3) has been developed which focuses on adding individual ant behavior. It has been applied in Capacitated Vehicle Routing Problem (CVRP). Therefore, this research aims to improve its solution by applying elitist ant concept, rearrange its selection of candidates and to include current status of vehicle capacity as part of its decision making. The result shows that DACS3 has achieve a better solution for most of the datasets. Applying elitist concept, rearrangement of candidate selection and vehicle capacity as part of its decision making can influence its achievement to reach an optimal distance.**

Keywords: Vehicle Routing, Ant Colony, Suitability Factor, Dynamic Candidate, Optimization.

I. INTRODUCTION

Ant Colony Optimization (ACO) algorithms have been applied in many type of Combinatorial Optimization Problem (COP). It has been proven to have a good algorithmic performance and enable to find good quality solution. Developing a good algorithm in helping to solve industrial based problem is always a challenge especially in logistic and transportation industries. Finding an efficient vehicle routes are important in maintaining industry competitive advantage. Reducing the length of delivery routes while decreasing the number of vehicle used enables businesses to provide better services to customers, having an efficient operations and possibly an increase in market share. This problem is important since time and cost associated to the fleet of vehicle in transporting products to geographically dispersed customers would determine the health of businesses financial. The more costs can be cut down resulted an increased in revenue, determine a healthy load of money that can be utilized for other purposes. Selecting a combination of customers would determine the selection of vehicle route for each vehicle. Distance is the main factor that contributes to the selection of customers. Thus it can be degraded into multiple Travelling Salesman Problem (TSP) processes. By having the problem capability which can be separated into sub-problem that can be solved using other method, it shows that VRP problem is a combinatorial optimization NP-hard problem where the number of feasible solutions for the problem increases exponentially with the number of customers to be serviced [1].

This paper is to show several improvement made to DACS3 which applied to CVRP. This paper is organized into several sections. Previous fundamental approaches and the case study were discussed in section two. Section three explains the algorithm used and the component that supports the algorithm operation. Section four shows the experimental setup while section five shows the experimental results and analysis. Lastly, the conclusion and discussion on further research are explained in section six.

II. PREVIOUS RESEARCH

Ant System (AS) was first introduced and applied to TSP by Marco Dorigo et. al. [2]. Initially, ants were placed on n cities and it moves from city r to city s using probabilistic formula called random proportional rules using Euclidean distance. After all ants have completed a tour, the pheromone level on all edges would be updated using local pheromone updating rule [3].

Later, Luca Maria Gambardella and his colleague [4] have modified the AS algorithm and introduced Ant Colony System (ACS) where it provides more balance and guidance in searching in three different ways. Firstly, the state transition rules (pseudo-random proportional action choice rules), provides a direct way to balance between an exploration of new edges and exploitation of a priori. Secondly, only edges that belong to the best ant tour being allowed to do pheromone

978-1-4244-6716-7/10/$26.00

updates through global pheromone updating rule and finally, local pheromone updating rule is applied while ants are trying to construct a solution [5]. Generally the basic idea was that the ants modified the pheromone level using local pheromone updating rule on the visited edges while constructing a solution after a series of choice selection on edges through decision making rule. After all ants have constructed a tour, it will then performed a second update using global pheromone updating rule following edges that belong to only one best solution which produces the shortest tour from the beginning of the trial [6].

Stutzle and Hoos [7] developed MAX-MIN Ant System (MMAS) which was a direct improvement of AS. They also consider some concept improvement made to ACS. MMAS is different in three ways [8]. The first improvement was only one ant would update the pheromone which is the model of ACS but it could choose whether to update on solution of the current iteration or following the global best solution. Secondly, the pheromone strength was to be bounded to upper and lower limit [t_{max}, t_{min}] in order to avoid search stagnation. Lastly, the initial value for pheromone strength was initializing to t_{max} which was intended to provide a higher search exploration of solution at the beginning of the algorithm runs. The basic idea was that by allowing ants to update the pheromone level considering the solution on iteration to iteration basis (preferred choice) would guarantee more pheromone activities which constitutes an improve of searching performance. Nonetheless, the initialization of the pheromone level to the highest would encourage more exploration activities but it was limited to its boundaries where it would ensure that pheromone information is limited to its trails and not allows it to be too intensified or completely lost.

Yi and Gong [9] have proposed an algorithm known as Dynamic Ant Colony System (DACS) which is a direct improvement of AS but considering improvement made to ACS and MMAS by introducing dynamic decay parameter to avoid the pheromone level growing too high and reaches local optima. With the theory that, pheromone evaporate quickly when it is intensified and less quickly when they are faint, the dynamic decay parameter was applied in both pheromone updates such as local pheromone updating rule and global pheromone updating rule. They are also trying to accelerate the solution computation by allowing the best and worst tour done by ants to do pheromone update.

The history of Vehicle Routing Problem (VRP) begins when Dantzig and Ramser introduced a real-world application concerning about the delivery of gasoline to the gas station back in 1959. It was where an algorithmic approach introduced through mathematical formulation in order to solve and produce solution to the problem. The work was then taken by Clarke & Wright when they inroduced an effective greedy heuristic algorithm that improved Dantzig and Ramser approach in 1964. Since then many models, algorithms and variances been proposed to find optimum or approximate solution for VRP. It has become a key ingredient and reference factor to many transportation systems. It is also among the famous combinatorial optimization problem because of its degree of difficulty and industry relevence [10] [11].

CVRP is considered as a foundation or basic of a complex problem in Vehicle Routing Problem (VRP). It is a little more complex problem to solve compared to TSP. For CVRP some restriction has been added. In addition to the famous TSP restriction that every customers can only be visited exactly once, all vehicle routes in CVRP must start and end at the depot and it cannot exceeds its vehicle capacity [12]. An individual ant simulates a vehicle. The route for the vehicle is constructed incrementally until all customers have been visited. All ants must start at the depot. It begins by selecting next customer to be served. The utilized capacity storage to serve that customer will be updated before the next customer is being selected. The ant or the vehicle will come back to the depot when vehicle capacity is exceeded or all capacity has been utilized. The ant will start a second tour if there are still customers to be served. The process continues when until there are no more customers to be visited and the total length of completed tours or routes will be calculated.

The CVRP problem description can be represented as a complete graph $G = (V, A)$ where $V = \{0,1,\dots,n\}$ is a set of vertex and $A = \{(i,j);\ i,j \in V; i \neq j\}$ is the set of arc. Vertices $V = \{1,\dots,n\}$ correspond to customers and vertex $V = \{0\}$ represented the depot. Sometimes the depot is associated with vertex n + 1. A nonnegetive cost of c_{ij} is associated with each arc $(i,j) \in A$, represents the travel cost of arc (i,j). A symmetric problem $c_{ij} = c_{ji}$ for all $(i,j) \in A$ and the arc set of A is commonly replaced by edges $E = \{(i,j); i,j \in V; i < j\}$. Let $d, R, m\ and\ Q$ denotes the customer demands, route of the vehicle, number of vehicles and the vehicle capacity respectively. For each vertex $i \in V$, an associated value $d_i \in \Re$ represents its demand. The total demand of all customers on a route must not exceed the vehicle capacity $Q_j \geq \sum_{i=1}^{m_j} d_{j_i}$. The solution of CVRP is the set of vectors which the sum costs of the tour $\sum_j \left(\sum_i c_{j_i}\right)$ and total number of vehicle utilization are minimal [13].

The attempt to solve VRP problem invites many approaches and ideas in ACO research community. The most famous approaches are by using saving algorithm in single colony approach or using multiple colonies of ants approach for multiple constraints or multiple objectives [14] [15] [16]. One similar feature in all approaches is that the solution constructed needs to be improved by local search. However in this paper, the comparison result is based on the concepts applied to CVRP. This will means that no local search heuristics will be considered.

III. DACS3 ON CVRP

A. DACS3 Algorithm

DACS3 algorithm considers the basic concepts introduced in ACS and DACS. Zulaiha and colleagues developed the concept of DACS3 based on the behavior found on Malaysian house red ants [17] [18] [19] [20]. DACS3 differs from previous systems in three ways. First, capturing all knowledge from the group and updating the pheromone level once the knowledge becomes available would expedite the process and increase the chances of finding a better solution. Second, a dynamic penalty on worst tours would open up chances for ants to navigate, limiting intentions and providing caution in an ant's decision to move. Finally, only one best tour from group performance is considered when applying the global pheromone updating rule, which is compared with two subdivided sections (best of the best and worst of the best).

In the local construction phase, all customers are considered as starting customer *r*. Every ant would have to make a complete tour without violating capacity constraint and the decision to choose which customer *s* to move to is provided by state transition rules in (1) and (2). Even though we use both exploitation and exploration decision rules to move, we favor exploration as the main strategy in finding solution. $d(i,j) = ([1 - \tau(i,j)] - [c(i) - d(i,j)])^{-1}$ is the suitability factor that represents the current vehicle capacity status *c(i)* subtracted by the demand of possible customer *d(i,j)* to be served. This will ensure the customer to be selected for the next move is suitable with the current capacity status of the vehicle. The inverse suitability factor would have pheromone influence $1 - \tau(i,j)$ as guidance.

$$s = \begin{cases} \arg\_max_{i \in J_k(i)} [\tau(i,j)].[n(i,j)]^{\beta}.[d(i,j)] & \text{If } q \le q_0 \\ P_{ij} & \text{Otherwise} \end{cases} \quad (1)$$

$$P_{ij} = \begin{cases} \dfrac{[\tau(i,j)].[n(i,j)]^{\beta}.[d(i,j)]}{\sum_{i \in J_k(i)} [\tau(i,j)].[n(i,j)]^{\beta}.[d(i,j)]^{1/2}} & \text{If i} \in J_k(i) \\ \text{i} & \text{Otherwise} \end{cases} \quad (2)$$

Every time an ant visits a customer j, it modifies the pheromone level by using the local pheromone updating rule [5]. The reason why we used the ACS version of local pheromone updating rule is because we believed that when an ant moves to a new location, it would make a constant pheromone deposit and evaporation.

$$\tau(r,s) \leftarrow (1 - [p.\tau(r,s)]).\tau(r,s) + \Delta C \quad (3)$$

Where

$$\Delta C = \begin{cases} p.\Delta\tau(i,j) & \text{if (r,s)} \in \text{Local Group Best Tour} \\ -\lambda[[p.\tau(i,j)].\Delta\tau(i,j)] & \text{if (r,s)} \in \text{Local Group Worst Tour} \\ 0 & \text{if (r,s)} \in \text{Others} \end{cases}$$

And

$$\Delta\tau(r,s) = \begin{cases} (L_{grb})^{-1} & \text{for Local Group Best Tour} \\ (L_{grw})^{-1} & \text{for Local Group Worst Tour} \end{cases}$$

Once all ants in the group completed their tours, the available knowledge of every member of the group will then be used to modify the pheromone level using the intermediate pheromone updating rule like (3) in the local reinforcement phase. The updates are necessary before all ants in the group can be given a new task to complete. In this phase, the dynamic decay parameter will be used because it helps to alleviate an early stagnation, reducing the possibility of pheromone levels growing too high. All current completed tours in the group will be compared to the group best tour in the current iteration and to the group worst tour from the beginning of trial. If no match is found, the edges would experience normal dynamic evaporation. This method will boost very effort the ants make to produce the best tour but dampen the worst tour from group performances. Dynamic penalty $[p.\tau(r,s)]$ is used to caution all ants off the bad paths on the next tour. L_{grb} is the total distance of the best tour in the current iteration and L_{grw} is the total distance of the worst tour of the group from the beginning of the trial.

$$\tau(r,s) \leftarrow (1 - [p.\tau(r,s)]).\tau(r,s) + \Delta C \quad (4)$$

Where

$$\Delta C = \begin{cases} \lambda.[p.\Delta\tau(i,j)] & \text{if (r,s)} \in \text{Global Best Tour} \\ -[p.\tau(i,j)].\Delta\tau(i,j) & \text{if (r,s)} \in \text{Global Worst Tour} \\ 0 & \text{if (r,s)} \in \text{Others} \end{cases}$$

And

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1} & \text{for Global Best Tour} \\ (L_{gw})^{-1} & \text{for Global Worst Tour} \end{cases}$$

The pheromone level is again modified using the global pheromone updating rule like (4) in global reinforcement phase. Only the best tour from the group performance will be considered for the pheromone updates. Every move in the solution or the tour will be compared with every move in the complete tours gathered for two categories, best of the best and worst of the best. This method will provide better search guidance in the effort to search for a better solution. L_{gb} is the total distance of the globally best tour (best of the best) and L_{gw} is the total distance of the globally worst tour (worst of the best) from the beginning of the trial.

B. Supporting Strategy

In improving DACS3 performance, we applied the candidate list strategy [4] [21] and an improvement idea on candidate list for VRP introduced by Bullnheimer et. al. [14]. The strategy has proved to help in selecting suitable search candidates and reducing algorithm computational time. However, its growth becomes unstable and unrealistic when involve large data. Another improvement by Zulaiha et. al. [22] called Dynamic Candidate List (DCL) has limit the growth and suitable to be used for all range of data. The equation for DCL is shown in (5). Short distance is always been the main selection criteria in determining candidates. It is a necessary element in fulfilling the objective function of best possible shortest solution. However, this approach might neglect those edges that receive high pheromone deposit or high desirability but not part of best solution. Not all edges that receive high pheromone deposit in a bad solution are a bad choice. It happened that the good edges was been selected as a part of a solution when an ant make a bad construction. This paper introduces two levels of selection criteria in DCL called Two Categories Dynamic Candidate List (2DCL) as shown in (6) in order to save good edges from completely forgotten by ants. This method will select and consider several edges which have high pheromone deposit or high desirability as part of candidate list selection. This means that one third of total candidate list will have pheromone as main selection criteria. 2DCL only commences where the main selecting criteria using distance as candidate selection completed.

$$DCL = \begin{cases} n/4 & \text{if } DCL \leq 30 \\ DCL^{*}/3 & \text{Otherwise until } DCL \leq 30 \end{cases} \tag{5}$$

Where DCL^{*} is previous value apply DCL.

$$2DCL = \frac{DCL}{3} \qquad \text{if } 2DCL < 1,\ 2DCL = 1 \tag{6}$$

Elitist strategy [23] is a concept to have a strong additional reinforcement to the edges that belong to the global best tour from the beginning of the trial. It was the first improvement proposed to Ant System [2]. An appropriate number of elitist ants allow the algorithm to find better solution early in the run. However, too many elitist ants would make the search reaches early stagnation. DACS3 applies this concept with a constant value of elitist ant called Lamda (**λ**). This elitist strategy was applied in DACS3 to global best tour shown in (4) and group worst tour shown in (3). It resembles a strong positive and negative reinforcement to appropriate edges.

Pheromone Trails Smoothing (PTS) is a repairing mechanism to stabilize the pheromone level to encourage more exploration introduced by Stutzle and Hoos [7]. The algorithms readjust the pheromone level on all edges by decreasing the pheromone level on edges that received a high pheromone influence and increases pheromone level on edges that received low pheromone influence. However, the weakness of this mechanism is the fact that it would affect the algorithm performance and reduce the ability of the algorithm in finding good quality solution if it is frequently used. DACS3 applied this repairing algorithm procedure but only used it when the algorithm reach a stagnant condition identified as *stagnant* = $n*2$ from the last found best solution. The result shows that the algorithm was able to help in finding good quality solution in less amount of time. The amount of pheromone whether increases or decreases depends on the parameter δ which is between 0 and 1. This paper adopted the setting (0.5) as used by Stutzle and Hoos.

IV. SIMULATION SETUP

The algorithm was tested using several datasets taken from VRP libraries [24] [25]. The results from these libraries are the compilation of the best known results found so far or the optimal solutions. The upper bound or the integer distance was taken as benchmark distances for the test comparison. The algorithm was developed using C language. Testing was performed on a machine with Intel® Pentium® M, 1.86GHz processor and 1 gigabyte of physical memory. The experiments sought to determine which algorithms between ACS and DACS3 could reach optimal distance with minimal vehicle utilization. If all tested algorithms were able to find it, then performance speed would be the second measurement.

For comparison as in Table 1, the first three columns are information about the case studies and it's currently best found so far results. The columns thereafter are set to be identical for the experimental results on the two tested algorithms. The first column is the best distance from the beginning of the trail, as compared to the benchmark distance. The second column shows the minimum vehicle utilization. The third column shows the number of iteration required to come up with the best distance. The fourth column is an average time taken from 15 trials.

Distance was measured by the integer distance (the roundup distance from each moves) and the real distance (in the bracket). The value is measured in Euclidean 2D and GEO distances. Real distance was used as a measurement in calculating distances for Euclidean datasets, while integer distance was used as the basis of the distance calculation for GEO datasets. The parameter settings for the DACS3 experiment on CVRP are as the same applied by Zulaiha et. al. [22].

TABLE I
Comparison Results between DACS3 and ACS

Problem Name	Benchmark Distance	Min Vehicle	DACS3				ACS			
			Best Distance	Min Vehicle	Best Iteration	Avg Time (Sec)	Best Distance	Min Vehicle	Best Iteration	Avg Time (Sec)
P-n16-k8	450 (N/A)	8	450 (451.95)	8	13	0.171	461 (462.69)	8	7705	60.235
B-n31-k5	672 (N/A)	5	672 (676.50)	5	6986	176.609	672 (676.30)	5	4138	165.406
B-n50-k8	1312 (N/A)	8	1355 (1358.49)	8	9357	899.594	1371 (1373.76)	8	9101	1510.860
P-n76-k4	593 (N/A)	4	623 (627.92)	4	4026	1124.719	673 (678.14)	4	339	188.343
P-n101-k4	681 (N/A)	4	734 (745.92)	4	8595	5692.172	792 (801.47)	4	2325	2999.375

V. RESULTS AND DISCUSSION

The result is captured in three forms. Table 1 shows the actual experimental results of DACS3 compared to the ACS on CVRP. The percentage difference between DACS3 and ACS to the benchmark distance is shown in Table 2. Fig. 1 to Fig. 5 shows the searching performance between DACS3 and ACS on different case studies. Generally, DACS3 is able to find better solution for most of the data except for B-n31-k5 data compare to ACS. However, the difference is very small around 0.02%. The results also show that the DACS3 basic principle of three level updates together with the suitability factor and its supporting strategies, can more effectively find solution up to 9%, in 4 out of 5 case studies based on table 2. For most of the data such as P-n16-k8, B-n31-k5, P-n76-k4 and P-n101-k4 datasets, DACS3 has performed well compared to ACS. However on B-n50-k8 data, DACS3 has a poor searching performance at the beginning but gradually improve and perform well in the middle till the end of the run.

TABLE II
Percentage Difference Between DACS3 and ACS to Benchmark

Problem Name	DACS3 (%)	ACS (%)
P-n16-k8	0.000	2.444
B-n31-k5	0.000	0.000
B-n50-k8	3.277	4.496
P-n76-k4	5.059	13.490
P-n101-k4	7.782	16.299

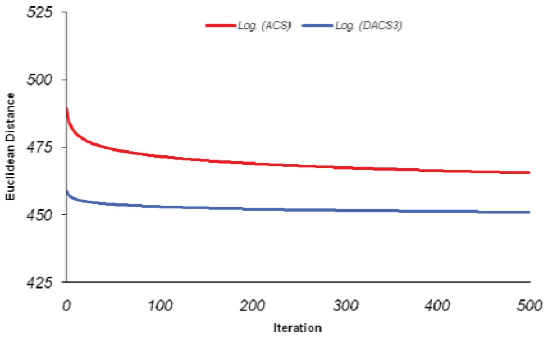

Fig. 1: Algorithm comparison graph for P-n16-k8 problem

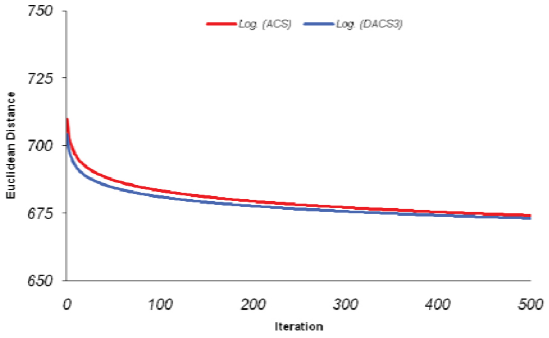

Fig. 2: Algorithm comparison graph for B-n31-k5 problem

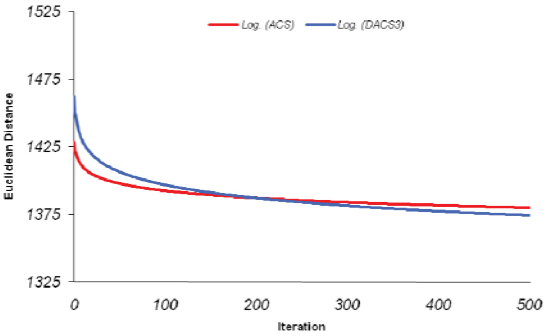

Fig. 3: Algorithm comparison graph for B-n50-k8 problem

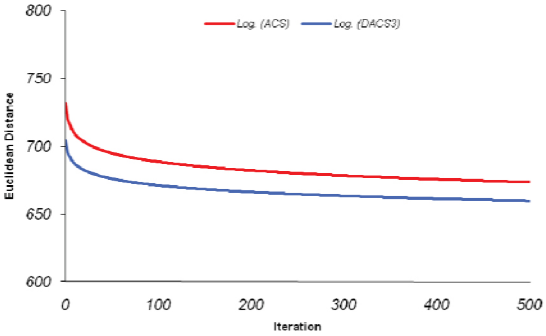

Fig. 4: Algorithm comparison graph for P-n76-k4 problem

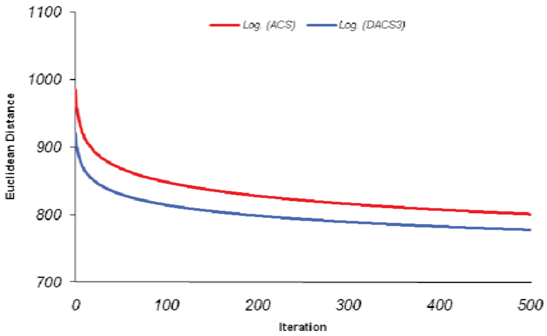

Fig. 5: Algorithm comparison graph for P-n101-k4 problem

VI. CONCLUSION AND FURTHER RESERACH

Embedding a simple single ant behavior projected in three level of updating rules model slows down the possibility of early stagnation. The addition of suitability factor which is the consideration of vehicle capacity as part of algorithm's decision making makes the algorithm achieved better searching capability. This capability will not work properly without the help of several supporting strategies such as a flexible selection of candidates, further intensification of pheromone and repairing mechanism. Thus, the following research is looking on capacity constraint again as part of its pheromone updating rules as further improvement and how DACS3 performed in larger data size as referred in [26].

REFERENCES

[1] Bell, J. E. & McMullen, P. R. 2004. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1): 41-48. Elsevier.

[2] Dorigo, M., Maniezzo V. & Colorni A. 1996. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics-Part B Cybernetics*, 26(1): 29-41. IEEE Systems, Man, & Cybernetics Society.

[3] Dorigo, M. & Caro, G. D. 1999. The Ant Colony Optimization Meta-Heuristic. In: *New Ideas in Optimization*, 11-32., McGraw-Hill.

[4] Dorigo, M. & Gambardella, L. M. 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1): 53-66, IEEE Computational Intelligence Society.

[5] Dorigo, M., & Socha, K. 2007. An Introduction to Ant Colony Optimization. *Book Chapter in Approximation Algorithms and Metaheuristic*. CRC Press.

[6] Dorigo, M., Birattari, M. & Stutzle T. 2006. Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. *IEEE Computational Intelligence Magazine*, 1(4): 28-39. IEEE Computational Intelligence Society.

[7] Stützle, T. & Hoos, H. H. 2000. MAX-MIN Ant system. *Journal of Future Generation Computer Systems*, 16(9): 889-914, Elsevier Science Publishers.

[8] Merloti, P. E. 2004. Optimization Algorithms Inspired by Biological Ants and Swarm Behavior. *Artificial Intelligence Technical Report CS550*. San Diego State University, San Diego.

[9] Li, Y. & Gong, S. 2003. Dynamic ant colony optimization for TSP. *International Journal of Advanced Manufacturing Technology*, 22(7): 528-533, Springer.

[10] Toth, P. & Vigo, D. 1987. The Vehicle Routing Problem. *Series: Monographs on Discrete Mathematics and Applications (No. 9)*. Philadelphia: SIAM.

[11] Maffioli, F. 2003. The Vehicle Routing Problem: A book review. *4OR: A Quarterly Journal of Operations Research*, 1(2): 149-153. Springer

[12] Bullnheimer, B., Hartl, R. F. & Strauss, C. 1999. An Improved Ant System Algorithm for the Vehicle Routing Problem. *Annals of Operations Research*, 89: 319-328. Springer.

[13] Lopes, H. S., Dalle Molle, V. L. & Erig Lima, C. R . 2005. An Ant Colony Optimization System for the Capacitated Vehicle Routing Problem. *Proceedings of the XXVI Iberian Latin-America Congress on Computational Methods in Engineering CILAMCE 2005*.

[14] Bullnheimer, B., Hartl, R. F. & Strauss, C. 1997. Applying the Ant System to the Vehicle Routing Problem. In I. Osman, S. Voss, S. Martello, and C. Roucairol (ed.). *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*, 109-120. France: Kluwer Academics.

[15] Doerner, K., Gronalt, M., Hartl, R. F., Reimann, M., Strauss, C. & Stummer, M. 2002. Savings Ants for the Vehicle Routing Problem. In S Cagnoni et. al. (ed.). *Lecture Notes in Computer Science: Application of Evolutionary Computing*, 2279: 73-109. Berlin: Springer.

[16] Gambardella, L. M., Taillard, T & Agazzi, G. 1999. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. Mcgraw-Hill'S *Advanced Topics In Computer Science Series: New ideas in optimization*. 63-76. McGraw-Hill, England, UK.

[17] Ali Othman, Z., Md Rais, H. & Hamdan, A. R. 2008. Embedding Malaysian House Red Ant Behavior into an Ant Colony System. *Journal of Computer Science,* 4(11): 934-941. Science Publication.

[18] Md Rais, H., Ali Othman, Z. & Hamdan, A. R. 2007. Improved Dynamic Ant Colony System (DACS) on symmetric Traveling Salesman Problem (TSP). *International Conference on Intelligent and Advanced System*, 43-48. IEEEXplore.

[19] Ali Othman, Z., Md Rais, H., Hamdan, A. R. 2008, DACS3: Embedding Malaysian Individual Ant Behavior in Ant Colony System. *Proceedings of World Academy of Science, Engineering and Technology*, 34: 527-532. Waset.org.

[20] Ali Othman, Z., Md Rais, H. & Hamdan, A. R.. 2009. DACS3: Embedding Malaysian Individual Ant Behavior in Ant Colony System. *International Journal of Intelligent Systems and Technologies*, 4(1): 65-70, World Academy of Science, Engineering and Technology (WASET.org).

[21] Md Rais, H., Ali Othman, Z., Hamdan, A. R. 2008. Reducing Iteration Using Candidate List. *International Symposium on Information Technology (ITSim2008)*, 3: 1-8. IEEEXplore.

[22] Ali Othman, Z., Md Rais, H., Hamdan, A. R. 2009. Strategies DACS3 Increasing its Performances. European Journal of Scientific Research, 27(4): 488-499. EuroJournals Publishing Inc.

[23] Dorigo M. and Stutzle T., 1999, ACO Algorithms for the Travelling Salesman Problem, Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategy, Evolutionary Programming, Genetic Programming and Industrial Applications, John Wiley & Sons.

[24] Vehicle Routing Datasets. http://branchandcut.org/VRP/data/#E. Accessed December 2009.

[25] The VRP Web. http://neo.lcc.uma.es/radi-aeb/WebVRP/. Accessed December 2009

[26] Golden, B. L. Vehicle Routing Problem Data. http://www.rhsmith.umd.edu/faculty/bgolden/vrp_data.htm. Accessed December 2009.