

Applying DACS3 in the Capacitated Vehicle Routing Problem

Helmi Md Rais¹, Zulaiha Ali Othman², Abdul Razak Hamdan³

*Computer and Information Science Department
Universiti Teknologi Petronas
Malaysia*

¹helmim@petronas.com.my

*Centre for Artificial Intelligence Technology
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
Malaysia*

²zao@ftsm.ukm.my

³arh@ftsm.ukm.my

Abstract ~ Ant Colony System (ACS) is a well known optimization algorithm to find a good route solution for logistics and transportation industries such as Traveling Salesman Problem (TSP) or Vehicle Routing Problem (VRP), for the company maximize the efficiency and resource. Several versions of Ant Colony Optimization (ACO) algorithms have been proposed which aim to achieve an optimum solution includes Dynamic Ant Colony System with Three Level Updates (DACs3). DACs3 is an enhancement of ACS which focuses on adding individual ant behavior. The algorithm works better in TSP solution. This research aims to see the performance of DACs3 in VRP domain. The result shows that DACs3 has achieved a better solution for most the datasets of Capacitated Vehicle Routing Problem (CVRP). Embedding a simple behavior of a single ant influences its achievement to reach an optimal distance and also can perform considerably faster compare to other algorithm in TSP and CVRP.

Keywords ~ Capacitated Vehicle Routing Problem (CVRP), Dynamic Ant Colony System with Three Level Updates (DACs3) and Ant Colony Optimization (ACO)

I. INTRODUCTION

Developing a good algorithm that helping 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.

Ant Colony algorithms has been proven to have a good algorithmic performance and enable to find good quality solution in many type of problems such as such as TSP, VRP [5] [6] [7], quadratic assignment problem (QAP) [8] [9], flow shop problem (FSP) [11], job shop problem (JSP) [12] [13] [14], sequential ordering problem (SOP) [15] [16], telecommunication network [17] [18], graph colouring problem [19] [20] [21], data mining [22] [23] [24] and etc.

Applying ACS in VRP is considered as multiple TSP process. It has problem capability which can be separated into sub-problem that can be solve using other method. VRP problem is a combinatorial optimization NP-hard problem where the number of feasible solutions for the problem increases exponentially with the number of customer to be serviced [33].

Our past research has developed an algorithm so called DACs3. It is an enhancement of ACS which focuses on adding individual ant behaviour. The basic concept of individual ant behaviour is the fact that each ant is provides with ability to update the pheromone level in three level update. The first update is same with the basic update in ACS, the second update is when after make decision of having best selection path among the good path. This method is similar with DACs [38]. While the third update is after make decision of having best selection path among the good path with value in first level update and among the best within the worse path. DACs3 has performed better in TSP compare with ACS and DACs [44]. The VRP is closely related to TSP but more challenging. The research described here builds on our success with the TSP

This paper is organized into several sections. Section two discusses the fundamental approaches in ant system, VRP and CVRP. Section three discusses more detail on DACs3 algorithm for CVRP, 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

A. ACO Background

Ant System (AS) was first introduced and applied to TSP by Marco Dorigo et. al. [25]. 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 [26].

Later, Luca Maria Gambardella and his colleague [27] 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 updates through global pheromone updating rule and finally, local pheromone updating rule is applied while ants are trying to construct a solution [28]. 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 [29].

Stutzle and Hoos [30] 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 [31]. 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. When MMAS is close to convergence, one mechanism called pheromone trail smoothing (PTS) was used that helps

increases pheromone trails proportionally to the maximum pheromone trails limit. The advantage of the mechanism is that, the information gathered during the run is not completely lost but merely weakened.

Yi and Gong [32] 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.

B. VRP Background

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 introduced 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 relevance [3] [4].

CVRP is considered as a foundation or basic of a complex problem in Vehicle Routing Problem (VRP). It is a little more difficult problem to solve compare 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 [2].

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 nonnegative 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 $d_i \in \mathfrak{R}$ represents its demand. The total demand of all customers on a route must not exceed the vehicle capacity $Q_j \geq \sum_{i=1}^{m_j} d_j$. The solution of CVRP is the

set of vectors which the sum costs of the tour $\sum_j (\sum_i c_{ji})$ and total number of vehicle utilization are minimal [1].

In VRP, an individual ant simulates a vehicle. The route for the vehicle is constructed incrementally until all customers have been visited. All ants start at the depot. It begins by selecting next customer to be served and 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 or when all customers have been visited or served. When the ant has utilized its capacity while there are still customers to be served, the ant will start a second tour. The process continues when until there are no more customers to be visited. If all customers have been visited, the total length of completed tours or route will be calculated.

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 [34] [35] [36]. One similar feature in all approaches is that the solution constructed needs to be improved by local search. Therefore, in this paper, the comparison result is based on the fundamental concepts applied to CVRP. This will mean that no local search 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 [37] [38] [39] [44]. 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). Fig. 1 shows the workflow of the DACS3 model.

In the local construction phase, all cities are considered as starting cities r . Every ant would have to make a complete tour without violating capacity constraint and the decision to choose which city s to move to is provided by state transition rules [27]. Even though we use both exploitation and exploration decision rules to move, we favor exploration as the main strategy in finding solution. Every time an ant visits a city s , it modifies the pheromone level by using the local pheromone updating rule [28]. 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.

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

Where

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

And

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

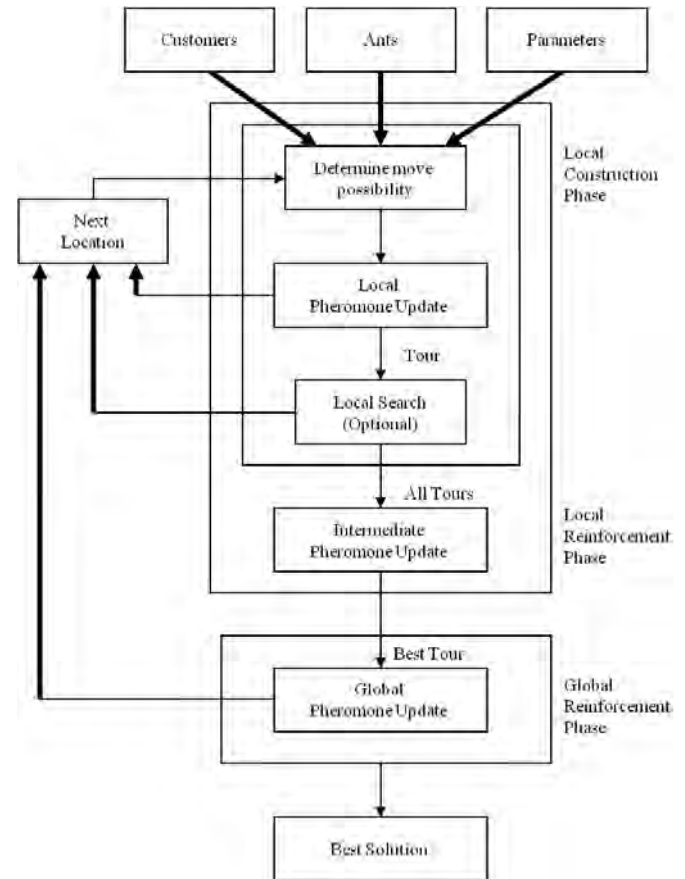


Fig 1: DACS3 model diagram

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 eq. 1 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 $(1 - [p \cdot \tau(r, s)])$ 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 arcs 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.

$$\tau(r, s) \leftarrow (1 - [p. \tau(r, s)]) \cdot \tau(r, s) + \Delta C \quad (2)$$

Where

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

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, eq. 2. 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. Fig. 2 shows how the DACS3 algorithm works. In this algorithm, we do not apply any local search heuristics.

STEP 1: {INITIALIZATION}

```
GlobalBestTour = ∞;
GlobalWorstTour = 0;
LocalGroupWorstTour = 0;
Generate initial solution using Nearest Neighbor (NN) heuristic;
Initialize pheromone level for all cities = τ0;
Determine Dynamic Candidate List (DCL);
CPU timer starts;
```

STEP 2: {TOUR CONSTRUCTION}

```
/* Trial begins */
Do
  /* Iteration begins */
  If i <= n (number of customer)
    LocalGroupBestTour = ∞;
    For k = 1 to m (number of ants)
      Load = capacity;
      Start city = 0;
      Next customer = i;
      Update current load;
      Update trail level customer (0,i);
```

```
Insert next customer i into tour list k;
Do
  STEP 2.1: {Moving Possibilities}
  Select the next customer j;
  STEP 2.2: {Local Pheromone Updates}
  /*Perform Local Pheromone Update*/
  If current loadj ≥ 0;
    Insert next customer j into tour list k;
    Update trail level customer (i,j);
    Update current load;
  Else
    Load = capacity;
    Next customer = 0 (depot);
  Endif;
  While (until all cities visited)
    Update trail level customer (j,0);
  EndFor
STEP 2.3: {Intermediate Pheromone Updates}
/*Perform Intermediate Pheromone Update*/
For k = 1 to m (number of ants / tours)
  Compute tour distance;
  If (tour distance < LocalGroupBestTour)
    LocalGroupBest = current solution;
  Else if (tour distance > LocalGroupWorstTour)
    LocalGroupWorst = current solution;
  Else /*Pheromone updates for others*/
    Update trail level for others;
  EndIf
  /*Pheromone updates for LocalGroupBest & LocalGroupWorst*/
  Update trail level for LocalGroupBest;
  Update trail level for LocalGroupWorst;
EndFor
EndIf
STEP 2.4: {Global Pheromone Updates}
/*Perform Global Pheromone Update*/
Compute tour distance of LocalGroupBest
If (tour distance < GlobalBestTour)
  Time taken for Global Best solution found so far;
  GlobalBest = LocalGroupBest;
Else if (tour distance > GlobalWorstTour)
  GlobalWorst = LocalGroupBest;
Else
  /*Pheromone updates for others*/
  Update trail level for others;
EndIf
/*Pheromone updates for GlobalBest & GlobalWorst*/
Update trail level for GlobalBest;
Update trail level for GlobalWorst;
Next customer (i) = i + 1;
STEP 3: {TERMINATING CONDITIONS}
While (until all termination statements satisfied);
```

Fig 2: DACS3 Algorithm for CVRP

B. Supporting Strategy

In improving DACS3 performance, we applied the candidate list strategy [27] [40] and an improvement idea on candidate list for VRP introduced by Bullnheimer et. al. [5]. 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. [41] 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 eq. 3 where n is the problem size

$$DCL = \begin{cases} \frac{n}{4} & \text{if } DCL \leq 30 \\ \frac{DCL^*}{3} & \text{Otherwise repeat until } DCL \leq 30 \end{cases} \quad (3)$$

Where DCL^* is previous value apply DCL

IV. EXPERIMENT SETUP

TABLE I
PARAMETERS for DACS3

Parameters	Values
Ant Population Size	10
q_0	0.8
β	5
p	0.6
Maximum Iterations	10000

The algorithm was tested using several datasets taken from VRP libraries [42] [43]. 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 2, 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 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 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. Table 1 shows the parameter settings for the DACS3 experiment for CVRP. q_0 is the relative importance of exploitation versus exploration, β is the relative importance of heuristic function and p is pheromone decay parameter.

V. RESULTS and DISCUSSION

The result is captured in three forms. Table 2 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 3. Figure 5a-e shows the searching performance between DACS3 and ACS. Generally, DACS3 is able to find better solution for most of the data except for B-n31-k5 data. As compared to ACS, DACS3 with basic principle of three level updates and its supportive strategy gives an improvement on the 'best distance' up to 2%, in 4 of the 5 cases. For most of the data, DACS3 has a poor searching performance at the beginning of the run but gradually improve and perform well in the middle till the end compare to ACS, which is shown in B-n50-k8, P-n76-k4 and P-n101-k4 datasets. However, DACS3 has an outstanding performance on P-n16-k8 but performing otherwise for B-n31-k5 datasets.

TABLE 2
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	26	0.235	461 (462.69)	8	7705	60.235
B-n31-k5	672 (N/A)	5	688 (691.01)	5	6044	114.000	672 (676.30)	5	4138	165.406
B-n50-k8	1312 (N/A)	8	1361 (1363.05)	8	3964	286.078	1371 (1373.76)	8	9101	1510.860
P-n76-k4	593 (N/A)	4	665 (673.10)	4	6766	1464.250	673 (678.14)	4	339	188.343
P-n101-k4	681 (N/A)	4	786 (795.69)	4	1007	519.734	792 (801.47)	4	2325	2999.375

TABLE 3
PERCENTAGE DIFFERENCE BETWEEN DACS3 and ACS to
BENCHMARK

Problem Name	DACS3 (%)	ACS (%)
P-n16-k8	0.000	2.444
B-n31-k5	2.380	0.000
B-n50-k8	3.734	4.496
P-n76-k4	12.141	13.490
P-n101-k4	15.418	16.299

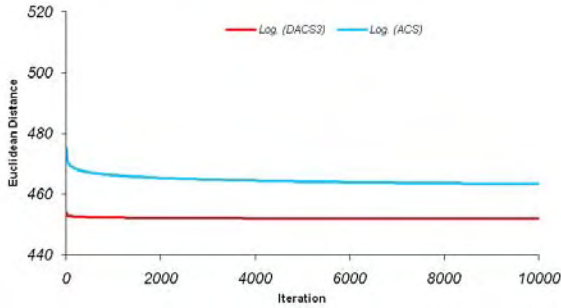


Fig 5a: Algorithm comparison graph for P-n16-k8 problem

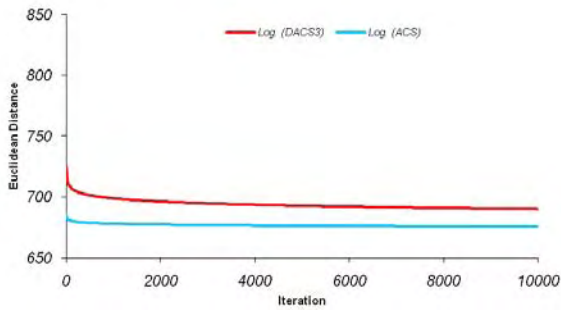


Fig 5b: Algorithm comparison graph for B-n31-k5 problem

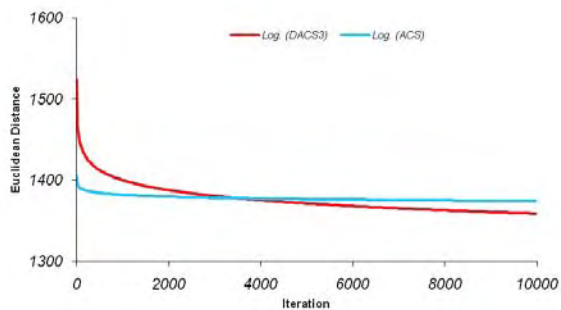


Fig 5c: Algorithm comparison graph for B-n50-k8 problem

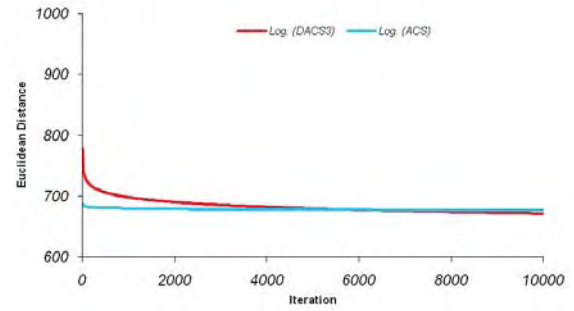


Fig 5d: Algorithm comparison graph for P-n76-k4 problem

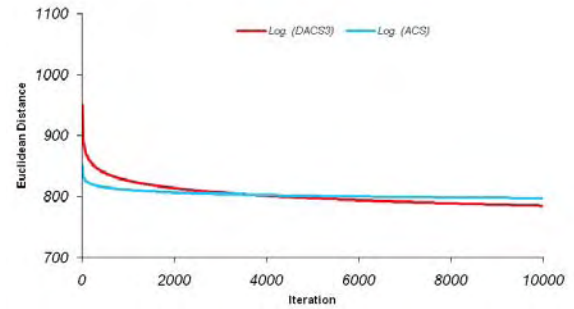


Fig 5e: Algorithm comparison graph for P-n101-k4 problem

VI. CONCLUSION and FURTHER RESEARCH

Embedding a simple single ant behavior projected in three level of updating rules or the model [13] slows down the possibility of early stagnation as shown in the comparison graphs. It also requires time to produce good solution due to influxation of information. However, the model has an improve possibility and capabilities of producing a good solution if ample time is allowed. Thus, the following research is looking the performance of DACS3 in large scale data set as stated in [45] and also looking on capacity constraint as part of its updating rules and decision making factor as further improvement.

REFERENCES

- [1] 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*.
- [2] 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.
- [3] Toth, P. & Vigo, D. 1987. The Vehicle Routing Problem. *Series: Monographs on Discrete Mathematics and Applications (No. 9)*. Philadelphia: SIAM.
- [4] Maffioli, F. 2003. The Vehicle Routing Problem: A book review. *4OR: A Quarterly Journal of Operations Research*, 1(2): 149-153. Springer
- [5] 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 Academic.
- [6] Applegate, D., Cook, W., Dash, S. & Rohe, A. 2002. Solution of a Min-Max Vehicle Routing Problem. *INFORMS Journal on Computing*, 14(2): 132-143. INFORMS.
- [7] Bouhafas, L., Hajjam, A. & Koukam, A. 2004. A Hybrid Ant Colony System Approach for the Capacitated Vehicle Routing Problem. In Marco Dorigo et. al. (ed.). *Lecture Notes in Computer Science: Ant Colony, Optimization and Swarm Intelligence*, 3172: 414-415. Berlin: Springer.
- [8] Gambardella, L. M., Taillard, E. D. & Dorigo, M.. 1999. Ant Colonies for the Quadratic Assignment Problem. *The Journal of the Operational Research Society*, 50(2): 167-176. Palgrave Macmillan.
- [9] Maniezzo, V. & Colomi, A. 1999. The Ant System Applied to the Quadratic Assignment Problem. *IEEE Transactions on Knowledge and Data Engineering*, 11(5): 769-778. IEEEExplore.
- [10] Randall, M. 2004. Near Parameter Free Ant Colony Optimisation. *Lecture Notes in Computer Science: Ant Colony, Optimization and Swarm Intelligence*, 3172: 374-381. Springer.
- [11] Stutzle, T. 1998. An ant approach to the flow shop problem. *Proceedings of the sixth European congress on intelligent techniques and soft computing (EUFIT' 98)*, 3: 1560-1564. Verlag Mainz, Aachen, Germany.
- [12] Zwaan, S. V. D. & Marques, C. 1999. Ant Colony Optimization for Job Shop Scheduling. *Proceedings of Third workshop on Genetic Algorithms and Artificial Life*.
- [13] Ventresca, M. & Ombuki, B. M. 2004. Ant Colony Optimization for Job Shop Scheduling Problem. *Technical Report # CS-04-04*. Brock University.
- [14] Zhang, J., Hu, X., Tan, X., Zhong, J. H. & Huang, Q. 2006. Implementation of an Ant Colony Optimization technique for job shop scheduling problem. *Transactions of the Institute of Measurement and Control*, 28(1): 93-108. SAGE.
- [15] Gambardella, L. M. & Dorigo, M. 1997. HAS-SOP: Hybrid Ant System for the Sequential Ordering Problem. *Technical Report IDSIA 11-97*. University of Lugano.
- [16] Gambardella, L. M. & Dorigo, M. 2000. An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem. *Informis Journal on Computing*, 12(3): 237-255. Informis.
- [17] Schoonderwoerd, R., Holland, O., Bruten, J. & Rothkrantz, L. 1996. Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 5(2): 169-207. MIT Press.
- [18] Bonabeau, E., Henaux, F., Guerin, S., Snyers, D., Kuntz, P. & Theraulaz, G.. 1998. Routing in telecommunications networks with ant-like agents. *Lecture Notes in Computer Science: Intelligent Agents for Telecommunication Applications*, 1437: 60-71. Springer.
- [19] Comellas, F. & Ozon, J. 1998. An Ant Algorithm for the Graph Colouring Problem. *Proceedings of the First International Workshop on Ant Colony Optimization*. 151-158.
- [20] Hertz, A. & Zufferey, N. 2006. A New Ant Algorithm for Graph Coloring. *Proceedings of the Workshop on Nature Inspired Cooperative Strategies for Optimization NICSO 2006*. 51-60.
- [21] Salari, E. & Eshghi, K. 2008. An ACO algorithm for the Graph Coloring Problem. *International Journal of Contemporary Mathematical Sciences*, 3(5-8): 305-310. Hikari.
- [22] Parpinelli, R. S., Lopes, H. S. & Freitas, A. A. 2001. An Ant Colony Based System for Data Mining: Applications to Medical Data. *Proceeding of International Conference of Knowledge Discovery and Data*, 55-62. Boston.
- [23] Parpinelli, R. S., Lopes, H. S., & Freitas, A. A. 2002. Data Mining with an Ant Colony Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*, 6(4): 321- 332. IEEEExplore.
- [24] Jiang, W., Xu, Y. & Xu, Y. 2005. A Novel Data Mining Method Based on Ant Colony Algorithm. *Lecture Notes in Computer Science: Advanced Data Mining and Applications*, 3584: 284-291. Springer.
- [25] Dorigo, M., Maniezzo V. & Colomi 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.
- [26] Dorigo, M. & Caro, G. D. 1999. The Ant Colony Optimization Meta-Heuristic. In: *New Ideas in Optimization*, 11-32., McGraw-Hill.
- [27] 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.
- [28] Dorigo, M., & Socha, K. 2007. An Introduction to Ant Colony Optimization. *Book Chapter in Approximation Algorithms and Metaheuristic*. CRC Press.
- [29] 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.
- [30] Stützle, T. & Hoos, H. H. 2000. MAX-MIN Ant system. *Journal of Future Generation Computer Systems*, 16(9): 889-914, Elsevier Science Publishers.
- [31] Merloti, P. E. 2004. Optimization Algorithms Inspired by Biological Ants and Swarm Behavior. *Artificial Intelligence Technical Report CS550*. San Diego State University, San Diego.
- [32] Li, Y. & Gong, S. 2003. Dynamic ant colony optimization for TSP. *International Journal of Advanced Manufacturing Technology*, 22(7): 528-533, Springer.
- [33] Bell, J. E. & McMullen, P. R. 2004. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1): 41-48. Elsevier.
- [34] 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 Academic.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] 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. IEEEExplore.
- [39] 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.
- [40] Md Rais, H., Ali Othman, Z., Hamdan, A. R. 2008. Reducing Iteration Using Candidate List. *International Symposium on Information Technology (ITSim2008)*, 3: 1-8. IEEEExplore.
- [41] 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.
- [42] Vehicle Routing Datasets. <http://branchandcut.org/VRP/data/#E>. Accessed December 2009.
- [43] The VRP Web. <http://neo.lcc.uma.es/radi-aeb/WebVRP/>. Accessed December 2009
- [44] 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).
- [45] Golden, B. L. Vehicle Routing Problem Data. http://www.rhsmith.umd.edu/faculty/bgolden/vrp_data.htm. Accessed December 2009.