Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm

Zar Chi Su Su Hlaing and May Aye Khine, *Member, IACSIT*

***Abstract*—Ant colony optimization (ACO) is a heuristic algorithm which has been proven a successful technique and applied to a number of combinatorial optimization problems and is taken as one of the high performance computing methods for Traveling salesman problem (TSP). TSP is one of the most famous combinatorial optimization (CO) problems and which has wide application background.. ACO has very good search capability for optimization problems, but it still remains a computational bottleneck that the ACO algorithm costs too much time to convergence and traps in local optima in order to find an optimal solution for TSP problems. The presented paper proposes an improved ant colony optimization algorithm with two highlights. First, candidate set strategy is adopted to rapid convergence speed. Second, a dynamic updating rule for heuristic parameter based on entropy to improve the performance in solving TSP. Algorithms are tested on benchmark problems from TSPLIB and test results are presented. From our experiments, the proposed algorithm has better performance than the conventional ACO algorithm and the results of the proposed algorithms are found to be satisfactory.**

***Index Terms*—Ant colony optimization, entropy, traveling salesman problem**

I. Introduction

Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behaviors of insects and of other animals. The attempt in the research of computer technology is to develop algorithms inspired by insect behavior to solve optimization problems. ant colony optimization (ACO) is one of the most successful techniques in the wider field of swarm intelligence. Many research works have been devoted to ant colony optimization techniques in different areas. It is a relatively novel meta-heuristic technique and has been successfully used in many applications especially problems that belong to the combinatorial optimization. ACO inspired by the foraging behavior of real ant was first introduced by dorigo and his colleagues ([1], [2]) in early 1990s and has become one of the most efficient algorithms for TSP. ACO is based on the pheromone trail laying and following behavior of some ant species, a behavior that was shown to allow real ant colonies to find shortest paths between their colony and food sources. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. The ants move according to the amount of pheromones, the richer the pheromone trail on a path is, the more likely it would be followed by other ants. So a shorter path has a higher amount of pheromone in probability, ants will tend to choose a shorter path. Artificial ants imitate the behavior of real ants how they forage the food, but can solve much more complicated problem than real ants can. Ant colony optimization exploits a similar mechanism for solving optimization problems.

From the early nineties, when the first ant colony optimization algorithm was proposed, ACO attracted the attention of increasing numbers of researchers and many successful applications are now available. ACO has been widely applied to solving various combinatorial optimization problems such as traveling salesman problem (TSP), job-shop scheduling problem (JSP), vehicle routing problem (VRP), quadratic assignment problem (QAP), Weapon-Target Assignment problems (WTA), etc.

ACO can be used to find the solutions of difficult combinatorial optimization problems and it enjoys a rapidly growing popularity. Although ACO has a powerful capacity to find out solutions to combinatorial optimization problems, it has the problems of stagnation and premature convergence and the convergence speed of ACO is always slow. Those problems will be more obvious when the problem size increases. Therefore, several extensions and improvements versions of the original ACO algorithm were introduced over the years. Various adaptations: an algorithm based on the basis of the ant evolution rules [3], dynamic control of solution construction and mergence of local search ([4], [5], [6]), new pheromone updating strategies [7], max-min ant system [8], a strategy is to partition artificial ants into two groups: scout ants and common ants [9], using candidate lists strategies ([10], [11]), dynamic ant colony system with three level updates ([12-13]) and using the path selection controlled by information entropy [15] are studied to improve the quality of the final solution and lead to speedup of the algorithm. All these studies have contributed to the improvement of the ACO to some extent, but they have little obvious effect on increasing the convergence speed and obtaining the global optimal solution. In the proposed system, the main modifications introduced by ACO are the following. First, ACO is more effective as the candidate set strategy is adopted. This modification reduces the size of the search space for the ant colony algorithm. Second, information entropy is introduced which is adjust the algorithm's parameters. Additionally, the best performing ACO algorithms for the TSP improve the solutions generated by the ants using local search algorithms. The experiment results show that the algorithm proposed in this study can substantially increase the convergence speed of the ACO.

Manuscript received November 17, 2011; revised November 30, 2011.
Authors are with the University of Computer Studies, Yangon, Myanmar (e-mail: zarchisusuhlaing@ gmail.com).

In this paper, a modified ant colony system for solving TSP using candidate set strategy and dynamic updating of heuristic parameter is developed. This algorithm is used to produce near-optimal solutions to the TSP. The paper is organized as follows: Section 2 describes traveling salesman problem. Section 3 and 4 illustrates the algorithm of ant colony system. Section 5 presents candidate list strategy approach to ACO and the other is analysis of heuristic parameter to be updated in the algorithm. Section 6presents the proposed algorithm for TSP. In Section 7, the proposed method is employed into several TSP problems and the results of our approach and of traditional ACO are reported. Finally, Section 8 makes the conclusion.

II. Traveling Salesman Problem

Traveling salesman problem (TSP) is a well known, popular and extensively studied problem in the field of combinatorial optimization and attracts computer scientists, mathematicians and others. Its statement is deceptively simple, but yet it remains one of the most challenging problems in operational research. It also an optimization problem of finding a shortest closed tour that visits all the given cities. It is known as a classical NP-complete problem, which has extremely large search spaces and is very difficult to solve.

The definition of a TSP is: given *N* cities, if a salesman starting from his home city is to visit each city exactly once and then return home, find the order of a tour such that the total distances (cost) traveled is minimum. Cost can be distance, time, money, energy, etc. TSP is an NP-hard problem and researchers especially mathematicians and scientists have been studying to develop efficient solving methods since 1950's. Because it is so easy to describe and so difficult to solve. Graph theory defines the problem as finding the Hamiltonian cycle with the least weight for a given complete weighted graph.

The traveling salesman problem is widespread in engineering applications. It has been employed in designing hardware devices and radio electronic devices, in communications, in the architecture of computational networks, etc. In addition, some industrial problems such as machine scheduling, cellular manufacturing and frequency assignment problems can be formulated as a TSP.

A complete weighted graph $G=(N, E)$ can be used to represent a TSP, where N is the set of n cities and E is the set of edges (paths) fully connecting all cities. Each $edge(i,j) \in E$ is assigned a cost d_{ij}, which is the distance between cities i and j. d_{ij} can be defined in the Euclidean space and is given as follows:

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \tag{1}$$

One direct solving method is to select the route which has minimum total cost for all possible permutations of *N* cities. The number of permutations can be very large for even 40 cities. Every tour is represented in 2n different ways (for symmetrical TSP). Since there are *n!* possible ways to permute n numbers, the size of the search space is then $|S|=n!/(2n)=(n-1)!/2$.

III. Theory and Mathematical Model of Ant Algorithm

A. The Theory of Ant Algorithm

The Ant Colony Optimization techniques has emerged recently as a relatively novel meta-heuristic for hard combinational optimization problems. It is designed to simulate the ability of ant colonies to determine shortest paths to food. Although individual ants posses few capabilities, their operation as a colony is capable of complex behavior.

Real ants can indirectly communicate by pheromone information without using visual cues and are capable of finding the shortest path between food sources and their nests. The ant deposits pheromone on the trail while walking, and the other ants follow the pheromone trails with some probability which are proportioned to the density of the pheromone. The more ants walk on a trail, the more pheromone is deposited on it and more and more ants follow the trail. Through this mechanism, ants will eventually find the shortest path. Artificial ants imitate the behavior of real ants how they forage the food, but can solve much more complicated problems than real ants can. A search algorithm with such concept is called Ant Colony Optimization. Figure 1 shows how the ants find the shortest path [18].

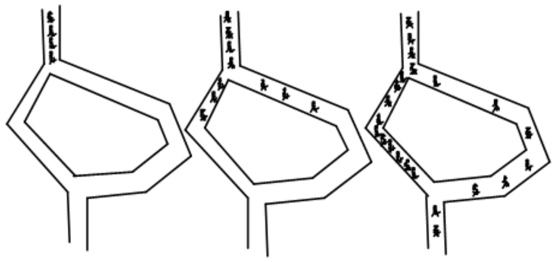

Fig. 1. Sketch map of the ant theory

B. Mathematical Model of Ant Algorithm

Ant System was first introduced and applied to TSP by marco dorigo et al. [13-15]. Initially, each ant is placed on some randomly chosen city. An ant *k* currently at city *i* chooses to move to city *j* by applying the following probabilistic transition rule:

$$p_{ij}^k(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^\alpha[\eta_{ij}]^\beta}{\sum_{l\in J_k(i)}[\tau_{il}(t)]^\alpha[\eta_{il}]^\beta} & if\ \ j\in J_k(i)\\ 0 & otherwise\end{cases} \tag{2}$$

where η_{ij} is the heuristic visibility of *edge (i, j)*, generally it is a value of $1/d_{ij}$, where d_{ij} is the distance between city *i* and city *j*. $J_k(i)$ is a set of cities which remain to be visited when the ant is at city *i*. α and β are are two adjustable positive parameters that control the relative weights of the pheromone trail and of the heuristic visibility. If α=0, the closed vertex is more likely to be selected. This is responding to a classical stochastic greedy algorithm. If on the contrary β=0, only pheromone amplification is at work: This method will lead the system to a stagnation situation, i.e. a situation in which all the ants generate a sub-optimal tour. So the trade-off between edge length and pheromone intensity appears to be necessary.

After each ant completes its tour, the pheromone amount on each path will be adjusted according to equation

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{3}$$

In this equation,

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{4}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_k} & \text{, if (i, j)} \in \text{tour done by ant k} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

(1-p) is the pheromone decay parameter (*0<p<1*) where it represents the trail evaporation when the ant chooses a city and decides to move. *m* is the number of ants, L_k is the length of the tour performed by ant *k* and Q is an arbitrary constant.

IV. Ant Colony System

Dorigo and other researchers have introduced many improved ACO algorithms based on AS, among which ant colony system (ACS) ([19], [20]) has better performance and is a representative of ACO. ACS mainly differs from AS in the way it chooses the next city to visit and in the way it updates pheromone levels on the edges. These changes increase the emphasis on exploitation by ensuring that most of the edges each ant follows occur around the best known tour.

Each ant is placed on some randomly chosen city as in AS. An ant *k* currently at city *i* choose to move to city *j* specified by the following rule:

$$j = \begin{cases} \arg\max_{u \in allowed_k(i)} \left\{ [\tau_{iu}]^{\alpha} . [\eta_{iu}]^{\beta} \right\} & \text{if } q < q_0 \\ J & \text{otherwise} \end{cases} \tag{6}$$

q_0 is a parameter that determines the importance of exploitation, $0 \le q_0 \le 1$ and q is a random number, $0 \le q \le 1$. $allowe_k(i)$ is the set of cities that have not been visited by ant *k*. *J* is a random variable determined in accordance with equation (2). This strategy obviously increases the variety of any searching, thus avoiding any premature falling into the local optimal solution and getting bogged down.

After each ant has chosen a city, the amount of pheromone on each side will be updated to equation (7) .

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho.\tau_0 \tag{7}$$

where ρ is a decay parameter $0 < \rho < 1$, $\tau_0 = 1/n.L_{nn}$ is the initial values of the pheromone trails, where n isthe number of cities in the TSP and L_{nn} is the cost produced by the nearest neighbor heuristic. A local pheromone updating rule encourages exploration of unused edges and avoids a local optimum by evaporating pheromone from the edges of each ant's tour. So, the effect of a local updating rule is to make an already edge less desirable for the following ant.

After all ants have constructed a tour, only the edges of the global best ant's tour from the beginning of the trail will be modified the pheromone level using the global pheromone updating rule

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho.\,\Delta\tau_{ij}(t) \tag{8}$$

$$\Delta\tau_{ij}(t) = \begin{cases} \dfrac{1}{L_{gb}} & \text{, if (i, j)} \in \textit{global best tour} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and L_{gb} is the length of the globally best tour found from the beginning of the trail.

The sketch of ACS can be shown in Fig.1.

```
Initialize
Loop /* at this level each loop is called an iteration */
    Each ant is positioned on a starting node
    Loop /* at this level each loop is called a step */
        Each ant applies a state transition rule to
        incrementally build a solution and
        A local pheromone updating rule
    Until all ants have built a complete solution
    A global pheromone updating rule is applied
Until end_condition
```

V. The Proposed Approach

A. Dynamic Candidate List Strategy

Candidate list is a strategy that tries to improve the performance of an ant algorithm. It was proposed by Gambardella and his colleague to accommodate searching procedure of ACS on larger data. The proposed candidate list is a dynamic candidate list procedure which captures a suitable number of nodes based on the total number of nodes. It is a static data structure that lists a limited number of preferred closed cities to be visited order by increasing distance. In the ACS algorithm, when the ant chooses the next city, the probability of its transfer from city *i* to city *j* needs to be computed, and then the city whose transfer probability (decision process) is first need to consider those preferred cities listed in the candidate list. Only when an ant cannot find suitable city to choose then the decision to choose a city will consider those which are outside the candidate list. The numbers of closest cities that allowed being included into the candidate list were different from one algorithm to another.

Due to the purpose of improving algorithm performances, the proposed system is also applying candidate list. However, it would not allow ants to venture into cities outside the candidate list. The number of cities or the size of the candidate list is also restricted to one fourth of the cities *n*. For example, seven was chosen resulting from the candidate list computation to determine the size of candidate list element for Oliver30 data. The candidate list procedure is as follows:

```
candidate_list=n/4 /*size of candidate list*/
determine cities that not yet visited
do
    for i=1 to n
      if city s is not yet visited
       determine distance between city r and city s
       if distance < distance of previous city s
      move city s into node_list
    end for
    candidate_list=node_list
while (until candidate_list is full)
```

B. Heuristic Parameter Updating

In ACO algorithm, the heuristic information is very important in generating high quality tours in the initial search stages. Because the value of the pheromone trails do not have much information in the early stage of learning and cannot guide the artificial ants in constructing good tours. In this situation, the heuristic parameter may be set to a large value. On the other hand, in the later stage, the heuristic parameter may need a small value because the pheromone trails may have collected enough information to behavior as required and the heuristic information may mislead the search due to its locality. Thus, in this situation, we may need a small value for the heuristic parameter. The heuristic parameter is set as a constant in traditional ACO algorithms. In this study, a high value of heuristic parameter can always provide high quality tours. This means that the influence of pheromone is greatly reduced, and ants are able to search other paths in constructing feasible solutions. It is evident that a small value of the heuristic parameter may result in bad performance in the early stage of learning. Nevertheless, a small value of the heuristic parameter can have good performance when the search process lasts long enough. Thus, it is intuitive to use an adaptive heuristic parameter for ACO. In this study, we intend to propose a way of designing an adaptive heuristic parameter for ACO such that the search performance can be better.

When ant colony algorithm begins to run, the amount of information on every path equals to each other, information entropy is maximum at this time, but as an enhancement of pheromone on the path, the entropy will be decreased gradually. If the entropy is not controlled currently, the entropy will eventually reduce to 0, that is, the pheromone on only one path is maximum, and the final solution will be mistaken, thus bringing about the premature. In order to overcome the easily-occurred precocious defects for solving complex combinatorial optimization problems with the basic ant colony algorithm, a proposed ant colony algorithm based on information entropy is discussed, using the heuristic parameter value selection controlled by information entropy.

Each trail is a discrete random variable in the pheromone matrix. The entropy of a random variable is defined as

$$E(X) = -\sum_{i=1}^{r} P_i \log P_i \tag{10}$$

where p_i represents the probability of occurrence of each trails in the pheromone matrix. For a symmetric n cities TSP, there are $n(n-1)/2$ distinct pheromone trails and $r= n(n-1)/2$. It is easy to see that when the probability of each trail is the same, E will be the maximum (denoted as E_{max}) and is given by

$$E_{\max} = -\sum_{i=1}^{r} P_i \log P_i = -\sum_{i=1}^{r} \frac{1}{r} \log \frac{1}{r} = \log r \tag{11}$$

We propose to use the entropy value as an index to indicate the degree about how much information has been learned into the pheromone trails and then the heuristic parameter can be updated accordingly. Notice that in this study, the heuristic parameter β is set to be an integer so as to avoid complicated computation because β is used as a power in Eqs. (2) and (6). Hence, we propose that β is update according to the rule given by

$$\beta = \begin{cases} 5 & threshold\ X < E' \le 1 \\ 4 & threshold\ Y < E' \le threshold\ X \\ 3 & threshold\ Z < E' \le threshold\ Y \\ 2 & 0 < E' \le threshold\ Z \end{cases} \tag{12}$$

$$E' = 1 - \frac{E_{\max} - E_{current}}{E_{\max}}$$

where E' is the entropy value for the current pheromone matrix and X, Y and Z are thresholds according to the city size. In study, threshold X is set within 0.8~0.9(according to the city size) and threshold B is within0.75~0.55 (according to the city size), and threshold Z is decided heuristically based on the value of Y.

VI. Proposed Algorithm

The proposed algorithm is combined with candidate list strategy and dynamic updating of heuristic parameter. The proposed algorithm is described as follows:

```
Procedure proposed ACO algorithm for TSP
Set parameters, initialize pheromone trails
Calculate the maximum entropy
Loop /* at this level each loop is called iteration*/
  Each ant is positioned on a starting node according to
  distribution strategy (each node has at least one ant)
  For k=1 to m do /*at this level each loop is called a step */
     At the first step moves each ant at different route
     Repeat
          Compute candidate list
          Select node j to be visited next (the next city in the
          candidate list) according to solution construction
          A local updating rule (7) is applied
     Until ant k has completed a tour
  End for
   Local search (2-opt, 2.5 opt) apply to improve tour
   A global updating rule (8) is applied
   Compute entropy value of current pheromone trails
   Update the heuristic parameter
Until end_condition
End
```

VII. Experimental Results

In order to describe the superiority of the proposed method, the algorithm was tested using several TSP problems. They are taken from the TSPLIB website [21]. In this study, we compared our proposed algorithm results with those of the ACS algorithm in the aspects of algorithm convergence and experiment results.

Table 1 presents the comparison of better results of tour length obtained from solving the TSP problems. In the proposed system, the parameters are set to the following values: $\rho = 0.1$, $q_0 = 0.7$, $\alpha=1$, β value is dynamically value of the proposed algorithm and $\beta=2$ is in ant colony system. The maximum iteration is set 20 times for all TSPs instances and the number of ants m is 20. The experiment shows that the ant colony algorithm proposed in this paper attained better results for TSPs, its efficiency of solution are higher than ant colony algorithm and the convergence speed is better than that of ant colony system.

TABLE I: Comparison of Tour Length Result of TSP Problems

Instance	Optimum (1)	Algorithm	Best (2)	Average	Relative error ((2)-(1))/(1)
berlin52	7542	ACS+2-opt	7542	7762.15	0%
		Proposed ACO	7542	7575.05	0%
eil51	426	ACS+2-opt	429	434.55	0.7%
		Proposed ACO	426	429.25	0%
eil76	538	ACS+2-opt	548	557.0	1.85%
		Proposed ACO	538	544.45	0%
st70	675	ACS+2-opt	677	688.95	0.29%
		Proposed ACO	675	683.85	0%
kroA100	21282	ACS+2-opt	21355	21790.65	0.34%
		Proposed ACO	21282	21423.25	0%
pr144	58537	ACS+2-opt	58620	58822.4	0.14%
		Proposed ACO	58537	58681.0	0%
lin105	14379	ACS+2-opt	14426	14745.1	0.32%
		Proposed ACO	14379	14472.6	0%

The following figures show the convergence comparisons. The solid line stands for the algorithm's convergence trait and the continuous dashed line stands for the ACS algorithm's trait. The experiment show that the convergence speeds of the proposed algorithm is obviously much faster than those of the ACS algorithm.

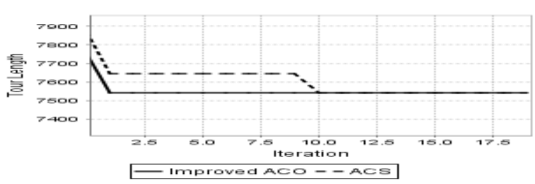

Fig. 2. Comparison of convergence speed of berlin52

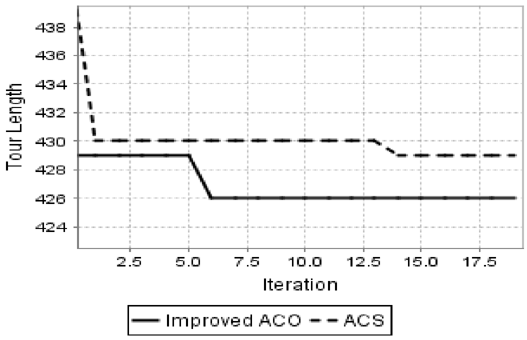

Fig. 3. Comparison of convergence speed of eil51

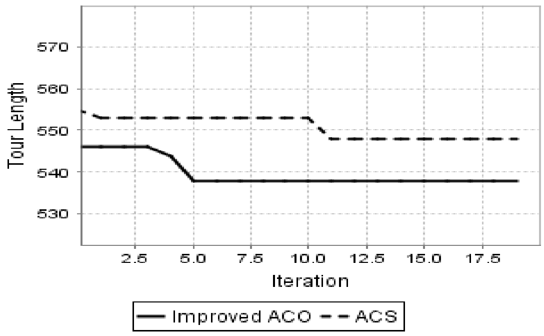

Fig. 4. Comparison of convergence speed of eil76

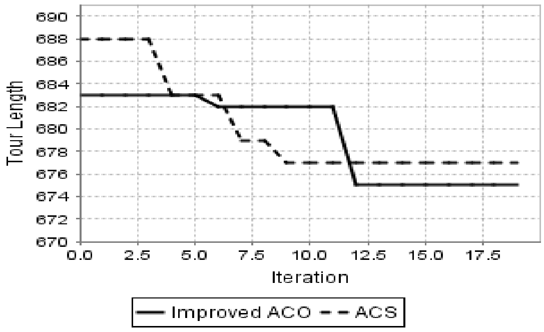

Fig. 5. Comparison of convergence speed of st70

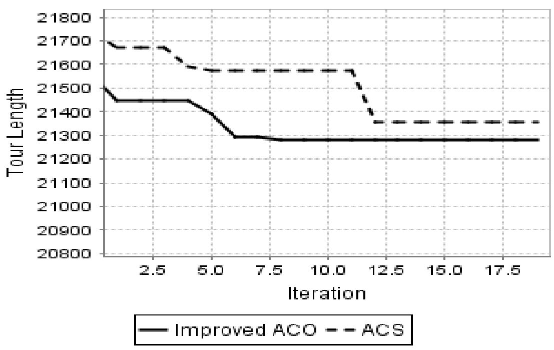

Fig. 6. Comparison of convergence speed of kroA100

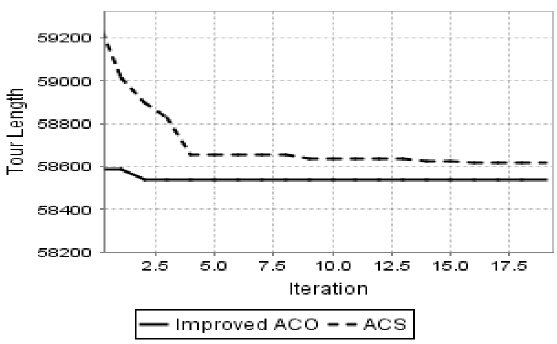

Fig. 7. Comparison of convergence speed of pr144 TSP

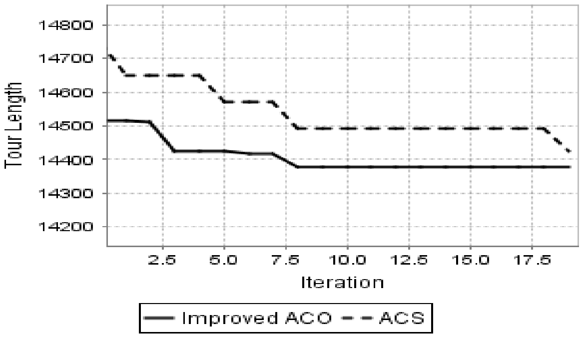

Fig. 8. Comparison of convergence speed of lin105

VIII. CONCLUSION

This paper presents an approach for solving traveling salesman problem based on improved ant colony algorithm. An improved version of ACO algorithm based on candidate list strategy and also proposed dynamic heuristic parameter updating based on entropy and mergence of local search solution is proposed. From our experimental results, the proposed system is more effective than the ACS algorithm in terms of convergence speed and the ability to finding better solutions.

ACKNOWLEDGMENT

I would like to express my heartfelt thanks to all my teachers for their valuable advice, helpful comments, kindness support, and precious time for my research. Most importantly, none of this would have been possible without the love and patience of my family throughout my Ph. D research. My heartfelt thanks also extend to all my colleagues and friends for their help, support, interest and valuable hints for discussions about research.

REFERENCES

[1] C. M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on System, Man, and Cybernetics*, Part B, vol.26, pp. 29-41, 1996.

[2] C.Blum, "Ant Colony Optimization: Introduction and recent trends," ScienceDirect, Physics of Life Reviews 2(2005)353-373.

[3] W. Tsai and F. Tsai, "A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules," *IJCNN 2002, IEEE.*

[4] H. Md. Rais, Z. A. Othman, and A. R. Hamdan, "Improved dynamic ant colony system (DACS) on symmetric Traveling Salesman Problem (TSP) ," *International Conference on Intelligence and Advanced Systems, IEEE*, 2007.

[5] J. Han and Y. Tian, "An improved ant colony optimization algorithm based on dynamic control of solution construction and mergence of local search solutions," *Fourth International Conference on Natural Computation*, IEEE, 2008.

[6] M. Colpan, "Solving geometric tsp with ants," the pennsylvania state university, 2005

[7] C.-M. Pintea and D. Dumitrescu, "Improving ant system using a local updating rule," *Proceedings of the Seventh International Symposium and Numeric Algorithms for Scientific Computing (SYNASC'05), IEEE* 2005.

[8] T. Stutzle and H. H. Hoos. "MAX-MIN ant system and local search for the traveling salesman problem," in *IEEE Int'l Conf. on Evolutionary Computation*. Indianapolis: IEEE Press, 1997.309~314.

[9] R. Gan, Q. Guo, H. Chang, and Y. Yi, "Improved ant colony optimization algorithm for the traveling salesman problems," *Jouranl of Systems Engineering and Electronics*, April 2010, pp 329-333.

[10] C.-X. Wang, D.-Wu. Cui, Y.-K. Zhang, and Z.-R. Wang, "A novel ant colony system based on delauney triangulation and self-adaptive mutation for tsp," *International Joural of Information Technology*, Vol.12, No.3, 2006.

[11] Z. A. Othman, H. Md. Rais, and A. R. Hamdan," Strategies DACS3 increasing its performances," *European Journal of Scientific Research*, 2009.

[12] Z. A. Othman, H. Md. Rais, and A. R. Hamdan, "DACS3: Embedding individual ant behavior in ant colony system," World Academy of Science Engineering and Technology 44 2008.

[13] H. Md. Rais, Z. A. Othman, and A. R. Hamdan, "Improvement DACS3 searching performance using local search," Conference on Data Mining and Optimization, IEEE, 27-28 October 2009.

[14] L. Li, S. Ju, and Y. Zhang, "Improved ant colony optimization for the traveling salesman problem", international conference on intelligent computation technology and automation, IEEE, 2008.

[15] X. Wei, "Improved ant colony algorithm based on information entropy," *International Conference on Computational and Information Sciences, IEEE,* 2010, 987-0-7695-4270-6.

[16] K. S. Hung, S. F. Su, and S. J. Lee, "Improving ant colony optimization for solving traveling salesman problem," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2007.

[17] D. X. Yu, "Hybrid ant colony optimization using memetic algorithm for traveling salesman problem," in *Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning* (ADPRD 2007).

[18] L. Min and J. Yant, "A shortest path routing based on ant algorithm," *Journal of Communication and Computer*, ISSN1548-7709, USA, September 2005.

[19] M. Dorigo and L. M. Gambardella, "The colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol.1, no.1, April, 1997.

[20] M. Dorigo, M. Birattari, and T. Stuzle, "Ant colony optimization, artificial ants as a computational intelligence technique," IEEE computational intelligence magazine, November 2006.

[21] TSPLIB WebPage, available:

[22] http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95 /tsp/

Zar Chi Su Su Hlaing was born in Amarapura, Mandalay Region, Myanmar, on 5th February 1980. She is an Assistance Lecturer of University of Computer Studies, Mandalay (UCSM), where she has been since 2008. She received a B.C.Sc., in 2003, B.C.Sc(Hons), in 2004, and a M.C.Sc, in 2007, from University of Computer Studies, Mandalay (UCSM). She joined her Ph.D. in Information Technology to the University of Computer Studies, Yangon (UCSY), in 2008, and now she is a Ph.D.(IT) (Thesis) research student. From 2005 to 2008 she worked at University of Computer Studies, Mandalay (UCSM), eventually as a Tutor.

Her research interests are in computational intelligence, swarm intelligence and intelligence informatics. She has explored the presence and implications of Combinatorial Optimization (CO) Problems.

Ms. Zar Chi Su Su Hlaing is an author of An Ant Colony Optimization for Solving Traveling Salesman Problem (International Conference on Information Communication and Management 2011), and An Approach for Solving Traveling Salesman Problem Using Hybrid Ant Colony Optimization (International Conference on Computer Applications 2011).