An Approach for Solving Traveling Salesman Problem using Hybrid Ant Colony Optimization

Zar Chi Su Su Hlaing, May Aye Khine
University of Computer Studies, Yangon
zarchissh@gmail.com, maya.khine@gmail.com

Abstract

*Traveling salesman problem (TSP) is one of the most famous combinatorial optimization (CO) problems, which has wide application background. 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 taken as one of the high performance computing methods for TSP. ACO has very good search capability for optimization problems, but it still has some drawbacks for solving TSP. These drawbacks will be more obvious when the problem size increases. The present paper proposes an ACO algorithm with nearest neighbor (NN) heuristic approach and information entropy which is conducted on the configuration strategy for the adjustable parameters to improve the efficiency of ACO in solving TSP. The performance of ACO also depends on the appropriate setting of parameters. Then, ACO for TSP has been improved by incorporating local optimization heuristic. Algorithms are tested on benchmark problems from TSPLIB and test results are presented. From our experiments, the proposed algorithm has superior search performance over traditional ACO algorithms do***.**

Keywords: ant colony optimization, traveling salesman problem, nearest neighbor heuristic

1. Introduction

Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behaviors of insects and of other animals. In particular, ants have inspired a number of methods and techniques among which the most studied and the most successful is the general purpose optimization technique known as ant colony optimization.

Ant Colony Optimization (ACO) is one of the most successful techniques in the wider field of swarm intelligence. ACO is inspired by 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), etc.

Although ACO has a powerful capacity to find out solutions to combinational optimization problems, it has the problems of stagnation and premature convergence and those problems will be more obvious when the problem size increases. In the proposed system, the main modifications introduced by ACO are the following. First, ACO is more effective if ants are initially visited thorough all cities according to the nearest neighbor (NN) approach to ACO to improve large TSPs thus obtain good solutions quickly. 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.

In this paper, an approach for solving TSP using hybrid ant colony optimization algorithm is developed. The paper is organized as follows: Section 2 is related work. Section 3 illustrates the algorithm of ant colony system. Section 4 describes traveling salesman problem. Section 5 presents nearest neighbor (NN) heuristic approach to ACO and the other is analysis of heuristic parameter to be updated in the algorithm. Section 6 presents 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.

2. Related Work

Cheng-Fa Tsai et al. [3] proposed an algorithm on the basic of the ant evolution rules. In addition, a method called nearest neighbor (NN) to EA to improve TSPs and obtain good solution quickly.

Kuo-Shen Hung et.al [4] proposed the analysis of lower pheromone trail bound and a dynamic updating rule for the heuristic parameters based on entropy to improve the efficiency of ACO. SHU Yunxing et al. [5] presented an ACO based on basic ACO algorithm on nearest neighbor node choosing rules and with crossover operator to increase the convergence speed of the ACO. Rongwei Gan et a.l [6] is to partition artificial ants into two groups: scout ants and common ants for solving the problems of basic ACO algorithm. The authors of reference [12] proposed an MMAS algorithm (max-min ant system) whose basic idea is to limit the pheromone density within the range of [min, max] so as to overcome the stagnation problem and meanwhile enhance the pheromone density along the path an individual ant has traveled in each iteration, thus increasing the convergence speed of the ACO.

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.

3. Ant Colony Optimization

The Ant Colony Optimization (ACO) techniques has emerged recently as a relatively novel meat-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.

ACO imitates the behavior of real ants to solve difficult combinatorial optimization problems. Real ants mark the paths that they walk on by laying down pheromone in a quantity. Other ants can observe the pheromone trail and are attracted to follow it. Because ants choose paths based on the amount of pheromone, the paths will marked again and will attract more ants. They are based on a colony of artificial ants that work cooperatively and communicate through artificial pheromone trails.

In each cycle (or iteration), all ants constructs a solution to the problem by travelling on a network. Each edge of the network represents the possible step that an ant can make and has associated two kinds of information that guide the ant's movement:

1. Heuristic information – measures the heuristic preference of moving from node i to node j. This information is not modified by the artificial ants during the algorithm run.

2. Pheromone trail information (artificial) – mimics the real pheromone that natural ants deposit. This information is modified during the

algorithm run depending on the solutions found by the ants.

In ACO, a number of artificial ants build solutions to the considered optimization problem at hand and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adapted by real ants. Skeleton of the ACO algorithm is as follows:

```
procedure ACO algorithm meta-heuristic
  Set parameters, initialize pheromone trails
   while (termination condition not met)   do
       ConstructSolutions
       ApplyLocalSearch % optional
       UpdateTrails
  End
end ACO algorithm meta-heuristic
```

ConstructSolutions: For the given problem instance, each ant starts a state, then traverses the states one by one.

At each step, each ant computes a probability distribution function to choose one of states in this probability. This random selection is called as random proportional transition rule.

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l\in J_k(i)}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} & if\ j\in J_k(i)\\ 0 & otherwise\end{cases} \qquad (1)$$

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 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.

And then, ACS also uses a transition rule, the pseudo-random proportional rule: an ant k positioned on city i chooses to move to neighbor node j by applying the rule given by eq.

$$j=\begin{cases}\arg\max_{l\in allowed_k(i)}\{\,[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}\,\} & \text{if } q<q_0\\ J & \text{otherwise}\end{cases} \qquad (2)$$

$q_0\in[0,1]$ is an initially set parameter; q is a random number and $q\in[0,1]$; J is a random variable determined in accordance with equation (3). This strategy obviously increases the variety of any searching, thus avoiding any premature falling into the local optimal solution and getting bogged down.

ApplyLocalSearch: Once solutions have been constructed, and before updating trail level, local search methods such as 2-opt, 2.5-opt and 3-opt can be applied on each solution constructed in current iteration. Although this process is optional and problem specific, it improves the solutions obtained by the ants and has been used by state of art ACO algorithms.

UpdateTrails: After the solutions constructed and calculated, pheromone level increases (called as pheromone depositing) and decreases (called as pheromone evaporation) on paths related to the good and bad solutions, respectively.

Local Trail Updating: This rule is applied each time an ant moves to a new city. The ant changes the trail on the edge will be updated according to the equation :

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \rho.\tau_0 \qquad (3)$$

where 0< ρ<= 1 is a decay parameter, $\tau_0=1/n.L_{nn}$ is the initial values of the pheromone trails, where n is the number of cities in the TSP and L_{nn} is the cost produced by the nearest neighbor heuristic. Eq. (1) is mainly to avoid very strong pheromone paths to be chosen by other ants and to increase the explorative probability for other paths. Once the edge between city i and city j has been visited by all ants, the local updating rule makes pheromone level diminish on the edge. So, the effect of the local updating rule is to make an already edge less desirable for a following ant.

Global Trail Updating: After all the ants have travelled through all the cities, update only the amount of the pheromone on the optimal path according to equation (6).

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \quad (4)$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{1}{L_k} & \text{, if the global best result is through path ij} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

4. TSP

Traveling salesman problem (TSP) is one of the well-known and extensively studied problems in combinatorial optimization. The goal is to find the shortest tour that allows each city to be visited once and only once.

The traveling salesman problem is also 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.

The definition of 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 totals distance (cost) traveled is minimum. TSP has extremely large search spaces and is very difficult to solve. In more formal terms, the goal is to find a Hamiltonian tour of minima length on a fully connected graph.

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)∈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:

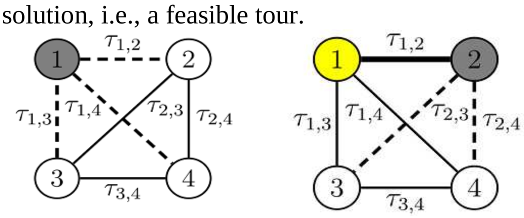
$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (6)$$

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 many 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 [9].

Moreover, an optimal solution to an instance of the TSP can be represented as a permutation π of the node (city) indices {1,2,……,n} such that the length *f*(π) is minimal, where *f* (π) is given by :

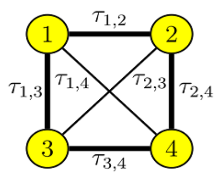
$$f(\pi) = \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)} \quad (7)$$

Concerning the ACO approach to TSP is described. The edges of the TSP graph can be considered solution components, i.e., for each $e_{i,j}$ is introduced a pheromone value $\tau_{i,j}$ and a heuristic information η_{ij}. The task of each ant consists of the construction of a feasible TSP solution, i.e., a feasible tour.



$$p(e_1,j) = \frac{\tau_{1,j}\eta_{1,j}}{\tau_{1,2}\eta_{1,2}+\tau_{1,3}\eta_{1,3}+\tau_{1,4}\eta_{1,4}} \qquad p(e_2,j) = \frac{\tau_{2,j}\eta_{2,j}}{\tau_{2,3}\eta_{2,3}+\tau_{2,4}\eta_{2,4}}$$

(a) First step **(b) Second step**



(c) The complete solution after the final construction step

Figure 1. Example of the solution construction for a TSP problem consisting of 4 nodes

The solution construction starts by randomly choosing a start node for the ant; in this case node 1. The current node of the ant is marked by dark gray color, and the already visited nodes are marked by yellow color (light gray color). The choices of the ant (i.e., the edges) are marked by dashed lines. The probabilities for the different choices (according to Eq. (1) or (2)) are given underneath the graphics. After the construction step, in which we exemplary assume the ant to have selected node 4, the ant can only move to node 3, and then back to node 1 in order to close the tour. This behavior will also be continued until best solution.

5. Proposed Approach

This paper presents a very powerful approach called improved ant colony optimization algorithm for TSP. Moreover, we represent a method called nearest neighbor (NN) heuristic to combine with ant colony system to improve search efficiency. The other is to analyze the effect of heuristic parameter required in ACO. Then, the system proposes a way of dynamically updating that parameter. This approach is based on the entropy measure of the pheromone in ACO.

5.1. Nearest Neighbor Heuristic Approach

The nearest neighbor heuristic for constructing a traveling salesman tour is the most simplest and straight forward. The nature of nearest neighbor heuristic is as follows:

Nearest Neighbor,

1. **Select** a random city.
2. **Find** the nearest unvisited city and go there.
3. Are there **any unvisited cities left**? If yes, **repeat step 2**.
4. **Repeat to the first city**.

In this paper we also add this concept to the proposed system for searching solutions space quickly. Consequently, we designed the nearest neighbor strategy in the computation attempt. The nearest neighbor (NN) tour can serve as good starting tours for subsequently performed improvement methods. Moreover, we utilize these near optimal solutions as the initial solutions for the proposed ant colony algorithm and thus reduce the convergence time. In other words, using nearest neighbor approach to obtain optimal solutions at initial stage will increase a little amount of pheromone trail strength at the ants searching paths and will reduce the searching time.

5.2. Analysis of Parameter Tuning

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.

The concept of entropy is known from Shannon's information theory. It is a measure of uncertainty concerning an event and is used to denote the degree of disorder in a system. Shannon's entropy represents the information regarding the probability of occurrence of an event. In ACO, pheromone is the basis of path selection, and the selection is uncertain in nature. Thus, we propose to consider the entropy information in ACO to estimate the variation of the pheromone matrix. Each trail is a discrete random variable in the pheromone matrix. The entropy of a random variable X is defined as

$$H(X) = -\sum_{t=1}^{r} P(x_t) \log P(x_t) \qquad (8)$$

where X represents the 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, H will be the maximum (denoted as H_{max}) and is given by

$$H_{max} = -\sum_{i=1}^{r} P_i \log P_i = -\sum_{i=1}^{r} \frac{1}{r} \log \frac{1}{r} = \log r \qquad (9)$$

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 tuned 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. (3) and (4). Hence, we propose that β is tuned according to the rule given by

$$\beta = \begin{cases} 5 & \text{threshold } X < H' \le 1 \\ 4 & \text{threshold } Y < H' \le \text{threshold } X \\ 3 & \text{threshold } Z < H' \le \text{threshold } Y \\ 2 & 0 < H' \le \text{threshold } Z \end{cases} \qquad (10)$$

where $H' = H_{current}/H_{max}$, $H_{current}$ is the entropy value for the current pheromone matrix and X, Y and Z are thresholds according to the city size. H′ will slowly decrease as the algorithm continues and the duration will be longer for large city numbers. In our study, threshold X is set within 0.8~0.9 (according to the city size) and threshold Y is within 0.75~0.55 (according to the city size), and threshold Z is decided heuristically based on the value of Y. In fact, our experiments show that no matter which values for those thresholds are used, as long as they are set properly.

6. Proposed Algorithm

The proposed algorithm is combined with nearest neighbor and entropy based dynamic heuristic parameter. The proposed algorithm is described as follows:

Procedure Proposed ACO algorithm for TSP
Set parameters, pheromone trails,
$\tau_{ij}(0) = \tau_0 = (nL_{nn})^{-1}$
Calculate the maximum entropy

$$H_{max} = -\sum_{i=1}^{r} P_i \log P_i = -\sum_{i=1}^{r} \frac{1}{r} \log \frac{1}{r} = \log r$$

Apply nearest neighbor approach
Loop /* at this level each loop is called iteration*/
Each ant is randomly placed on starting node
For k=1 to m (m=no: of ants)**do /*at this level each loop is called a step */**
At the first step moves each ant at different route
Repeat
Select node j to be visited next (the next node must not be visited by the ant)
A local updating rule equation [3] is applied
Until ant *k* has completed a tour
End for
Local search apply to improve tour (2-opt or 2.5- opt)
The following global updating rule is applied

$$\tau_{ij}(t+1) \begin{cases} (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) & \text{if(i, j)} \in \text{global - best - tour} \\ \tau_{ij}(t) & \text{otherwise} \end{cases}$$

$$\Delta\tau_{ij}(t) = \frac{1}{L_{gb}}$$

Compute entropy value of current pheromone trails with equation (8)
Update the heuristic parameter with equation(11)
Until End_condition
End

7. Experimental Results

In order to demonstrate the superiority of the proposed method, eil51 TSP problem is considered. It is obtained from the TSPLIB website. In this study, we compared our results with those of the ACS algorithm and those of the proposed algorithm in convergence and experiment results.

The following figures present the comparison of better results obtained from solving the problem. In the proposed system, the parameters are set to the following values: $\rho=\psi=0.1$, m=10 (ant number), $q_0=0.7$, $\alpha=1$, β value is dynamically value of the proposed algorithm. The maximum iteration is set 20 times. The experiment shows that the ant colony algorithm proposed in this paper attained results for TSPs, its efficiency of solution are higher than ant colony algorithm which is the improvement of the basic ant algorithm.

Table 1 presents the comparison of the better results of tour length obtained from solving the eil51 problem. In Table 1, Standard deviation=(solution-TSPLIB solution)/TSPLIB solution shows in parenthesis and "-" shows that it has no standard deviation (i.e, solution means best, average and worst solution). Figure 1, 2, 3 and 4 shows the test results for eil51 problem; best solution, average solution and worst solution of the tour lengths of the algorithms. These figures show the tour length results of using the dynamic β value of the proposed algorithm and the various results of ant colony algorithm of using the value of β=2, 3, 4 and 5 respectively. The solution of proposed algorithm are better that that of ant colony algorithm.

Table 1. Analysis of tour length results of TSP eil51 instance

eil51		Best solution	Average solution	Worst solution
ACO+2opt	β=2	428 (0.46%)	436.05 (2.35%)	443 (3.99%)
	β =3	430 (0.93%)	435.65 (2.26%)	443 (3.99%)
	β =4	429 (0.7%)	436.0 (2.34%)	443 (3.99%)
	β =5	429 (0.7%)	434.7 (2.04%)	440 (3.28%)
Proposed algorithm X=0.9, Y=0.7, Z=0.6		426 (-)	429.25 (0.76%)	432 (1.4%)
The best solution from TSPLIB: 426				

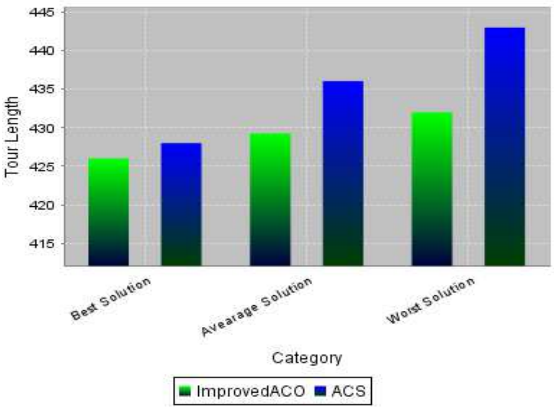


Figure 2. Comparison of tour length of eil51 problem of proposed algorithm with dynamic β value and ACO algorithm with β=2

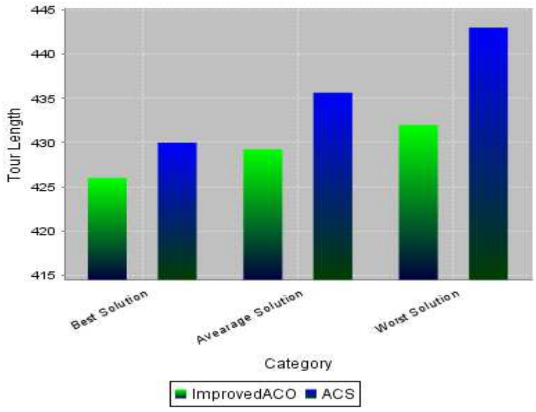


Figure 3. Comparison of tour length of eil51 problem of proposed algorithm with dynamic β value and ACO algorithm with β=3

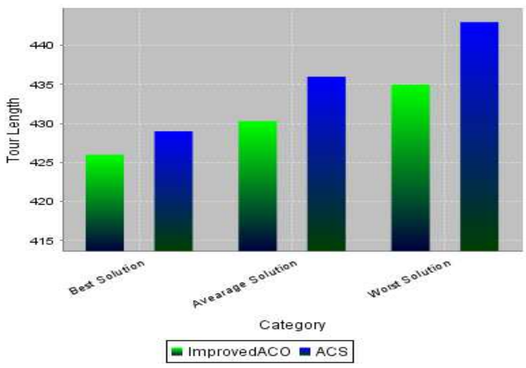


Figure 4. Comparison of tour length of eil51 problem of proposed algorithm with dynamic β value and ACO algorithm with β=4

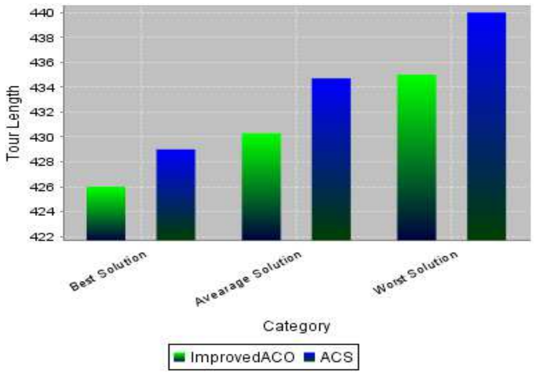


Figure 5. Comparison of tour length of eil51 problem of proposed algorithm with dynamic β value and ACO algorithm with β=5

8. Conclusion

This paper proposed an improved ant colony algorithm to improve search efficiency for solving TSP. In the initiation, the system presents a method called nearest neighbor (NN) to ant colony algorithm to improve TSP solutions quickly and such tour can guide ants toward the effective solution space in the initial search stages. Moreover, the entropy rule is introduced to adjust algorithm's parameters. It can also be found that the proposed dynamic update of heuristic parameter based on entropy will produce high better tours. Then, local search applied tour improvement. So, the proposed system reaches the superior search performance over traditional ACO algorithms do.

References

[1] 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.

[2] 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.

[3] C. F. Tsai, C. W. Tsai. "A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules", In: Neural Networks, IJCNN 2002, Proc.of the 2002 Int'l Joint Conf. on, Vol 2.Honolulu: IEEE Press,2002.1540~1545.

[4] K. S. Hung, S.F. Su, S. J. Lee, "Improving Ant Colony Optimization for Solving Traveling Salesman Problem", Jouranl of Advanced Computational Intelligence and Intelligent Informatics, 2007.

[5] S. Yunxing, G. Junen, G. Bo, " An Ant Optimization based on the Nearest Neighbor Node Choosing Rules and Crossover Operator", IEEE Computer Society, 2008.

[6] R. Gan, Q. Guo, H. Chang, Y. Yi, " Improved Ant Colony Optimization Algorithm for the Traveling Salesman Problems", Jouranl of Systems Engineering and Electronics, April 2010, pp 329-333.

[7] G.Ma, H.Duan, S.Liu, " Improved Ant Colony Algorithm for Golbal Optimal Trajectory Planning of UAV under Complex Environment", International Journal of Computer Science and Applications, Vol4 Issue 3, pp57-68.

[8] L.Min, J. Yant, " A Shortest Path Routing Based on Ant Algorithm", Journal of Communication and Computer, ISSN1548-7709, USA, September 2005, Volume 2, No.9.

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

[10] A. Ugur, D. Aydin, "An Interactive Simulation and Analysis Software for Solving TSP using Ant Colony Optimization Algorithms", ScienceDirect, Advances in Engineering Software 40 (2009) 341-349.

[11] T. White, S. Kaegi, T. Oda, "Revisiting Elitism in Ant Colony Optimization", School of Computer Science, Carleton University, 2003.

[12] TSPLIB WebPage, http://www.iwr.uniheidelberg.de/groups/comopt/software/ TSPLIB95/tsp/.

[13] T. Stutzle, 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.

[14] H. Duan, X. Yu, "Hybrid Ant Colony Optimization Using Memetic Algorithm for Traveling Salesman Problem", Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRD 2007).

[15] X-s. Yan, H-m. Liu, J. Yan, Q-h. Wu, "A Fast Evolutionary Algorithm for Traveling Salesman Proble", Third International Conference on Natural Computation, IEEE 2007.

[16] M. Dorigo, M. Birattari, T. Stuzle, "Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique", IEEE Computational Intelligence Magazine, November 2006.