*2011 International Conference on Information Communication and Management*
*IPCSIT vol.16 (2011) © (2011) IACSIT Press, Singapore*

# An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem

Zar Chi Su Su Hlaing, May Aye Khine

University of Computer Studies, Yangon

**Abstract.** Ant Colony Optimization (ACO) is a heuristic algorithm which has been proven a successful technique and applied to a number of combinatorial optimization (CO) problems. The traveling salesman problem (TSP) is one of the most important combinatorial problems. ACO is taken as one of the high performance computing methods for TSP. It still has some drawbacks such as stagnation behavior, long computational time, and premature convergence problem of the basic ACO algorithm on TSP. Those problems will be more obvious when the considered problems size increases. The proposed system based on basic ACO algorithm with well distribution strategy and information entropy which is conducted on the configuration strategy for updating the heuristic parameter in ACO to improve the performance in solving TSP. 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 better performance than ACO algorithm.

**Keywords:** ant colony optimization, traveling salesman problem, entropy

## 1. Introduction

In recent years, many research works have been devoted to ant colony optimization (ACO) techniques in different areas. It is a relatively novel meta-heuristic technique and has been successfully used in many applications especially problems in combinatorial optimization. ACO algorithm models the behavior of real ant colonies in establishing the shortest path between food sources and nests. Ants can communicate with one another through chemicals called pheromones in their immediate environment. The ants release pheromone on the ground while walking from their nest to food and then go back to the nest. 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. Through this mechanism, ants will eventually find the shortest path. Artificial ants imitate the behavior of real ants, but can solve much more complicated problem than real ants can.

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 the convergence speed of ACO is very 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: dynamic control of solution construction [4], mergence of local search [3, 13], a strategy is to partition artificial ants into two groups: scout ants and common ants [11] and new pheromone updating strategies [1, 3, 14], using candidate lists strategies [2, 16, 17] 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 extents, 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, to avoid search stagnation and ACO is more effective if ants are initially placed on different cities. 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 improved ant colony optimization algorithm is developed for solving TSP. 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 illustrates the background theory of ant colony system. Section 4 presents distribution strategy of initial ants and analysis of heuristic parameter to be updated in the algorithm. In Section 5, the proposed method is employed into several TSP problems and the results of our approach and of traditional ACO are reported. Finally, Section 6 makes the conclusion.

## 2. Traveling Salesman Problem

Traveling salesman problem (TSP) is one of the well-known and extensively studied problems in discrete or combinational optimization and asks for the shortest roundtrip of minimal total cost visiting each given city (node) exactly once. TSP is an NP-hard problem and 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. It is widespread in engineering applications and 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)∈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:

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

## 3. ACO Background

### 3.1. Ant System

Ant System was first introduced and applied to TSP by Marco Dorigo et al. [7, 8, and 9]. Initially, each ant is randomly put on a city. During the construction of a feasible solution, ants select the following city to be visited through *a probabilistic decision rule*. When an ant *k* states in city *i* and constructs the partial solution, the probability moving to the next city *j* neighboring on city *i* is given by

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum\limits_{u \in J_k(i)} [\tau_{iu}(t)]^\alpha [\eta_{iu}]^\beta} & \mathit{if} \;\; j \in J_k(i) \\ 0 & \mathit{otherwise} \end{cases} \tag{2}$$

where $\tau_{ij}$ is the intensity of trails between edge (i.j) and $\eta_{ij}$ is the heuristic visibility of edge (i, j), and $\eta_{ij}=1/d_{ij}$. $J_k(i)$ 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. After each ant completes its tour, the pheromone amount on each path will be adjusted with equation (3).

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

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- ρ)* is the pheromone decay parameter *(0<ρ<1)* where it represents the trail evaporation when the ant chooses a city and decide to move. $L_k$ is the length of the tour performed by ant *k* and *m* is the number of ants.

### 3.2. The ACS Algorithm

The ACS is mainly different from the AS in these aspects: The decision rules of the ants are different; the global updating rules are different; and local updating rules which adjust the amount of the pheromone on various paths are newly added.

Step 1: Initiation. The amount of the pheromone on each side is initiated into a tiny constant value; allocate *m* ants randomly to *n* cities.

Step 2: In ACS, the so-called *pseudorandom proportional rule* is used: the probability for an ant to move from city *i* to city *j* depends on a random variable *q* uniformly distributed over [0, 1], and a predefined parameter $q_0$.

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

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.

Step 3: The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the chosen city, $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho.\tau_0$ (7)

where $0< \rho<= 1$ 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. equation (2) 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.

Step 4: Computing of the optimal path. After m ants have travelled through all the cities, compute the length of the optimal.

Step 5: Global updating of pheromone. After all the ants have travelled through all the cities, update only the amount of the pheromone on the optimal path with equation (8):

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

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

where $\rho$ is constant and $L_{gb}$ is the length of global best tour.

Step 6: If the designated search number is not attained, then repeat the above steps.

# 4. Proposed Approach

## 4.1. Well Distribution Strategy of Initial Ants

At the beginning of ACO algorithm, some paths are walked through by the ants, others are never passed. The local heuristic controlled by visibility, encourages them to choose cities which are closer. This means that they are likely to choose to travel along short edges. Thus some cities may have many ants while some cities may have no ant at all. Because the amount of pheromone on each path is initially identical, therefore the ant mainly uses the distance between the two cities as the heuristic factor when it chooses the next city. In this way, when there are relatively more ants in a certain city, the density of the pheromone on a certain path will be strengthened due to the relatively larger number of ants travelling along the path. The path is not necessarily the shortest path that local optimum solution is searched out or ants arrive at stagnation state.

In order to solve this problem, the system adopted a method to distribute the ants evenly, i.e., position m ants to n cities and make sure that each city receives at least one ant. Thus, the search space of the solution is enlarged and the probability of getting the best result is increased.

### 4.2. Heuristic Parameter Updating

In ACO algorithm, the heuristic information is very important in generating high quality tours. Because the value of the pheromone trails do not get enough information in the early stage of learning and cannot guide the ants in constructing good tours, 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 to local optimal solution. The heuristic parameter is set as a constant in traditional ACO algorithms. In this study, if a heuristic parameter value is big, the ant will seek for other route found now, which will cause the premature. 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 updating controlled by 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. The maximum entropy 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 the rule given by

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

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

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.

### 4.3. Proposed Algorithm

The proposed algorithm is combined well distribution strategy of initial ants 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**
      Select node j to be visited next (the next node must not be visited by the ant) according to
      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**

## 5. Experimental Results

In order to validate the efficiency of the proposed method, several TSP problems are considered. They are obtained from the TSPLIB [19]. In this study, we compared its performance with the ACS algorithm. In all experiments, parameters are set to the following values: $\rho$= 0.1, $q_0$ = 0.7, $\alpha$=1, $\beta$ value is dynamically value of the proposed algorithm and $\beta$=2 is in ant colony system. The maximum iteration is set 20 times.

In order to compare the proposed algorithm with reference [14] and [15], some TSP instances are used. A comparison of final solution is shown in Table. 1 (the results of reference [14] and [15] are directly taken from these papers). Table 2 presents the comparison of better results obtained from solving the TSP problems. The experiment shows that the proposed algorithm is more effective than the conventional ACO in terms of convergence speed and the ability to finding better solutions.

**Table 1. Comparison of tour length results of TSP problems**

| TSP | Best length of proposed algorithm | Best length of reference [14] | Best length of reference[15] |
|---|---|---|---|
| eil51 | 426 | - | 429.98 |
| eil76 | 538 | 548.2376 | - |
| berlin52 | 7542 | 7544.3659 | - |
| st70 | 675 | 677.1076 | 677.1096 |

**Table 2. A comparison between proposed algorithm and ACS**

| TSP | Optimum (1) | Proposed algorithm Best (2) | Average | Relative error ((2)-(1))/(1) | ACS+2-opt Best (3) | Average | Relative error ((3)-(1))/(1) |
|---|---|---|---|---|---|---|---|
| kroA100 | 21282 | 21282 | 21384.2 | 0% | 21379 | 21756.4 | 0.46% |
| kroA150 | 26524 | 26524 | 27142.1 | 0% | 27249 | 27756.3 | 2.73% |
| pr144 | 58537 | 58537 | 58637.75 | 0% | 58603 | 58809.15 | 0.11% |

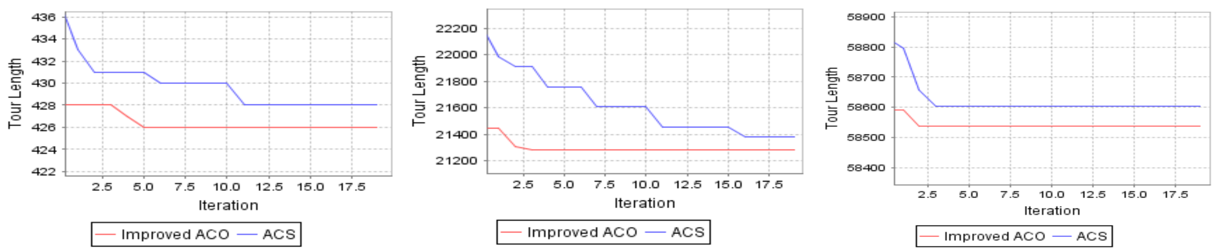

Fig. 1: Comparison of tour length for eil51, kroA100 and pr144 TSP Problems

## 6. Conclusion

This paper presents an approach for solving traveling salesman problem based on improved ant colony algorithm. The main contribution of this paper is a study of the avoidance of stagnation behavior and premature convergence by using distribution strategy of initial ants and dynamic heuristic parameter updating based on entropy. Then a mergence of local search solution is provided. The experimental results and performance comparison showed that the proposed system reaches the better search performance over ACO algorithms do.

## 7. References

[1] C-Mihaela Pintea, 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.

[2] H. Md. Rais, Z. A. Othman, A.R. Hamdan, *Reducing Iteration Using Candidate List*, IEEE, 2008.

[3] H. Md. Rais, Z. A. Othman, A.R. Hamdan, *Improvement DACS3 Searching Performance using Local Search*, Conference on Data Mining and Optimization, IEEE, 27-28 October 2009.

[4] J. Han, 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.

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

[6] L. Li, S. Ju, Y. Zhang, *Improved Ant Colony Optimization for the Traveling Salesman Problem*, International Conference on Intelligent Computation Technology and Automation, IEEE, 2008.

[7] 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.

[8] 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.

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

[10] Y.Li, *A Novel ACO with Average Entropy*, Journal of Software Engineering and Applications, Science Research, 2009, 2: 370-374.

[11] 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.

[12] TSPLIB WebPage, http://www2.iwr.uni-heidelberg.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] X-song, B. LI, H. YANG, *Improved Ant Colony Algorithm and Its Applications in TSP*, Proceedings of Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), IEEE, 2006.

[15] Y. Zhang, Z-l. Pei, J-h. Yang, Y-c. Liang, *An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Traveling Salesman Problem*, IEEE 2007. 1-4244-1509-8.

[16] Z. A. Othman, H. Md. Rais, A.R. Hamdan, *DACS3: Embedding Individual Ant Behavior in Ant Colony System*, World Academy of Science Engineering and Technology 44 2008.

[17] Z. A. Othman, H. Md. Rais, A.R. Hamdan, *Strategies DACS3 Increasing its Performances*, European Journal of Scientific Research, 2009.