ISSN 2250-3153

An Ant Colony Optimization Algorithm for Solving Travelling Salesman Problem

Krishna H. Hingrajiya, Ravindra Kumar Gupta, Gajendra Singh Chandel

University of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M. P.)

Abstract- Ant colony optimization (ACO) has been widely used for different combinatorial optimization problems. In this paper, we investigate ACO algorithms with respect to their runtime behavior for the traveling salesperson (TSP) problem. 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. There are several reasons for the choice of the TSP as the problem to explain the working of ACO algorithms it is easily understandable, so that the algorithm behavior is not obscured by too many technicalities; and it is a standard test bed for new algorithmic ideas as a good performance on the TSP is often taken as a proof of their usefulness.

Index Terms- Ant colony optimization, Traveling salesman problem

I. Introduction

In the case of ACO algorithms the theoretical analyses of their runtime behavior has been started only recently.We increase the theoretical understanding of ACO algorithms by investigating their runtime behavior on the well-known traveling salesperson (TSP) problem. For ACO the TSP problem is the first problem where this kind of algorithms has been applied. Therefore, it seems to be natural to study the behavior of ACO algorithms for the TSP problem from a theoretical point of view in a rigorous manner. ACO algorithms are inspired by the behavior of ants to search for a shortest path between their nest and a common source of food. It has been observed that ants find such a path very quickly by using indirect communication via pheromones. This observed behavior is put into an algorithmic framework by considering artificial ants that construct solutions for a given problem by carrying out random walks on a so-called construction graph. The random walk (and the resulting solution) depends on pheromone values that are values on the edges of the construction graph. The probability of traversing a certain edge depends on its pheromone value.

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.

Consider Fig. 1A Ants arrive at a decision point in which they have to decide whether to turn left or right. [12,13] Since they have no clue about which is the best choice, they choose randomly. It can be expected that, on average, half of the ants decide to turn left and the other half to turn right. This happens both to ants moving from left to right (those whose name begins with an L) and to those moving from right to left (name begins with a R). Figs. 1B and 1C show what happens in the immediately following instants, supposing all ants walk at approximately the same speed. The number of dashed lines is roughly proportional to the amount of pheromone that the ants have deposited on the ground. Since the lower path is shorter than the upper one, more ants will visit it on average, and therefore pheromone accumulates faster. After a short transitory period the difference in the amount of pheromone on the two path is sufficiently large so as to influence the decision of new ants coming into the system (this is shown by Fig. 1D). From now on, new ants will prefer in probability to choose the lower path, since at the decision point they perceive a greater amount of pheromone on the lower path. This in turn increases, with a positive feedback effect, the number of ants choosing the lower, and shorter, path. Very soon all ants will be using the shorter path.

The travelling salesman problem (TSP) is the problem of finding a shortest closed tour which visits all the cities in a given set. In this article we will restrict attention to TSPs in which cities are on a plane and a path (edge) exists between each pair of cities (i.e., the TSP graph is completely connected).

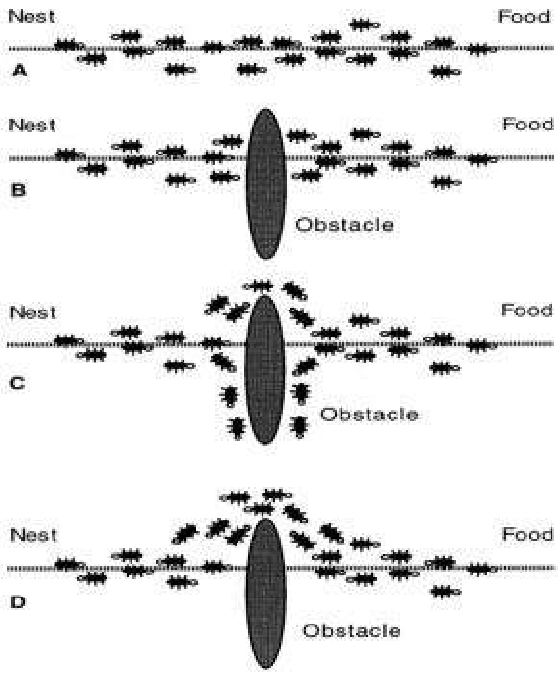


Fig.1: how real ant finds the shortest path

A) Ants arrive at a decision point. B) Some ants choose the upper path and some the lower path. The choice is random. C) Since ants move at approximately constant speed, the ants which choose the lower, shorter, path reach the opposite decision point faster than those which choose the upper, longer, path. D) Pheromone accumulates at a higher rate on the shorter path. The number of dashed lines is approximately proportional to the amount of pheromone deposited by ants [12,13].

The most interesting aspect of this autocatalytic process is that finding the shortest path around the obstacle seems to be an emergent property of the interaction between the obstacle shape and ants distributed behavior: although all ants move at approximately the same speed and deposit a pheromone trail at approximately the same rate, it is a fact that it takes longer to contour obstacles on their longer side than on their shorter side which makes the pheromone trail accumulate quicker on the shorter side. It is the ants preference for higher pheromone trail levels which makes this accumulation still quicker on the shorter path. We will now show how a similar process can be put to work in a simulated world inhabited by artificial ants that try to solve the travelling salesman problem.

The above behavior of real ants has inspired ant system, an algorithm in which a set of artificial ants cooperate to the solution of a problem by exchanging information via pheromone deposited on graph edges. Ant system has been applied to combinatorial optimization problems such as the traveling salesman problem.

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.

II. TRAVELLING SALESMAN PROBLEM

The traveling salesman problem (TSP) is the problem of finding a shortest closed tour which visits all the cities in a given set. In this article we will restrict attention to TSPs in which cities are on a plane and a path (edge) exists between each pair of cities (i.e., the TSP graph is completely connected) [12,13].

Traveling salesman problem (TSP) is one of the well-known and extensively studied problems indiscrete 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. 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..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 *dij*, which is the distance between cities i and j. *dij* can be defined in the Euclidean space and is given as follows:

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

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 40cities. 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. ACO MODEL

3.1 The ANT System

Ant System was first introduced and applied to TSP by marcodorigo et al. Initially, each ant is placed on some randomly chosen city. An ant k currently at city I choose to move to city j by applying the following probabilistic transition rule:

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

where nij is the heuristic visibility of edge (i, j), generally it is a value of 1/dij, where dij is the distance between city i and city j. Jk (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 I 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 benecessary.After each ant completes its tour, the pheromone amount on each path will be adjusted according to equation *(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, *Lk* is the length of the tour performed by ant *k* and Q is an arbitrary constant [12,13].

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

In this equation,

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

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

2.2 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 q0.

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

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,

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

Where 0< ρ<= 1 is a decay parameter, τ0=1/n.Lnn is the initial values of the pheromone trails, where n is the number of cities in the TSP and Lnn 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):

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

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

Where ρ is constant and Lgb is the length of global best tour.

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

IV. 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 mants 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 in the initial search stages. Because the value of the pheromone trails do

ISSN 2250-3153

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 toots 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 bebetter.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, theentropy 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

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

Where *pi* 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 *Emax*) and is given by

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

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,

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

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

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.

V. 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

VI. Experimental Results

In order to validate the efficiency of the proposed method, several TSP problems are considered. They are obtained from the TSPLIB. In this study, we compared its performance with the ACS algorithm. In all experiments, parameters are set to the following values: ρ= 0.1, q0 = 0.7, α=1, β value is dynamically value of the proposed algorithm and β=2 is in ant colony system. The maximum

International Journal of Scientific and Research Publications, Volume 2, Issue 8, August 2012

ISSN 2250-3153

iteration is set 20 times. In order to compare the proposed algorithm with reference [1] and [2], some TSP instances are used.

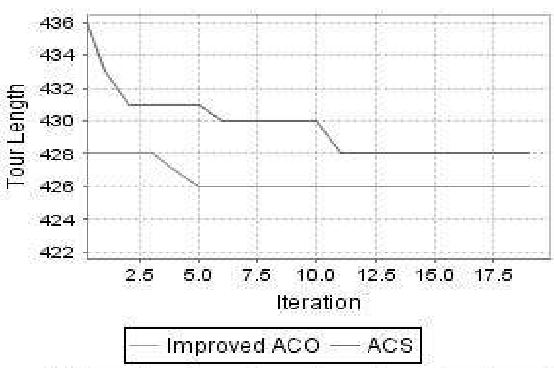
A comparison of final solution is shown in Table. 1 (the results of reference [1] and [2] are directly taken from these papers). Table 2 presents the comparison of better results obtained from solving the TSP problems. The experiment shows that the proposed algorithm is more effective than the conventional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

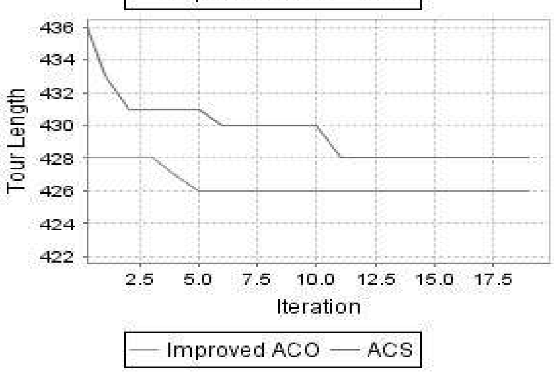
Table I Comparison of tour length results of TSP problems

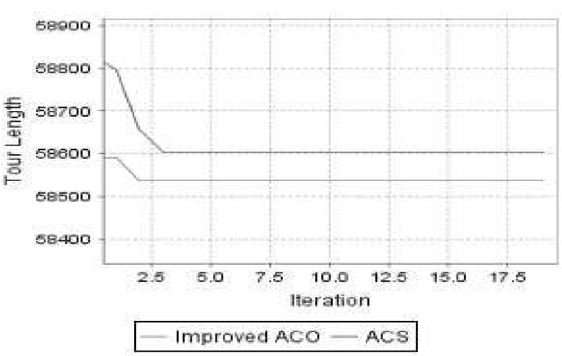
TSP	Best length of proposed algorithm	Best length of reference[1]	Best length of reference[2]
eil51	426	-	429.98
eil76	538	548.2376	-
bBerlin52	7542	7544.3659	-
st70	675	677.1076	677.1096

Table II A comparisons between proposed algorithm and ACS

TSP	Optimum (1)	Best (2)	Average	Relative Error ((2)-(1)/(1)	Best (3)	ACS +2 opt 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%
Pr1444	58537	58537	58637.75	0%	58603	58809.15	0.11%







VII. 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 emergence 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. The proposed system is more in terms of convergence speed and the ability to finding better solutions.

REFERENCES

[1] 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.

[2] 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.

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

[4] 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.

[5] M. Dorigo and M. Gambardella. Ant colonies for the travelling salesman problem. BioSystems 43 (1997) 73–81

[6] 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

[7] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies,"*Proceedings of ECAL91* - European Conference on

Artificial Life, Paris, France, 1991, F.Varela and P. Bourgine (Eds.), Elsevier Publishing, pp. 134–142.

[8] M. Dorigo, V. Maniezzo, and A.Colorni, "The ant system: optimization by a colony of coop-erating agents," IEEE Transactions on Systems, Man, and Cybernetics–Part B, vol. 26, No. 2, pp. 29–41, 1996.

[9] V. Maniezzo, A.Colorni, and M.Dorigo, "The ant system applied to the quadratic assignmentproblem, *Tech Rep. IRIDIA/94-28*, 1994, UniversitéLibre de Bruxelles, Belgium.

[10] Zar Chi Su SuHlaing and May Aye Khine, "Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm", International Journal of Information and Education Technology, Vol. 1, No. 5, December 2011

[11] [C. Horoba and D. Sudholt. Running time analysis of ACO systems for shortest path problems. In T. St¨utzle,M. Birattari, and H. H. Hoos, editors, SLS, volume 5752 of Lecture Notes in Computer Science, pages 76–91. Springer, 2009.

[12] M. Dorigo, L M Gambardella Ant colonies for the traveling salesman problem. BioSystems, 1997

[13] M. Dorigo, L M Gambardella Ant Colony system; A cooperative learning approach to the Travelling salesman problem. IEEE Transactions on Evolutionary Computation, Vol.1, No.1, 1997

AUTHORS

First Author – Krishna H. Hingrajiya, University of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M. P.), 1krishnahingrajiya@gmail.com

Second Author – Kumar Gupta, University of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M. P.)

Third Author – Gajendra Singh Chandel, University of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M. P.)