Ant Colony Optimization (ACO) For The Traveling Salesman Problem (TSP) Using Partitioning

Alok Bajpai, Raghav Yadav

Abstract: An ant colony optimization is a technique which was introduced in 1990's and which can be applied to a variety of discrete (combinatorial) optimization problem and to continuous optimization. The ACO algorithm is simulated with the foraging behavior of the real ants to find the incremental solution constructions and to realize a pheromone laying-and-following mechanism. This pheromone is the indirect communication among the ants. In this paper we introduces the partitioning technique based on the divide and conquer strategy for the *traveling salesman problem (*which is one of the most important combinatorial problem) in which the original problem is partitioned into the group of sub problems. And then we apply the ant colony algorithm using candidate list strategy for each smaller sub problems. After that by applying the local optimization and combining the sub problems to find the good solution for the original problem by improving the exploration efficiency of the ants. At the end of this paper we have also be presented the comparison of result with the normal ant colony system for finding the optimal solution to the traveling salesman problem.

Index Terms: Ant Colony Optimization (ACO), Mathematical Model of ACS, Traveling Salesman Problem (TSP), stigmergic communication, Candidate List, local optimization, Partitioning.

1 INTRODUCTION

The ant colony optimization first introduced in 1991 by A. Colorni M. Dorigo, and V. Mahiezzo [3]. The algorithm is simulated with the behavior of the real ants. The real ants having the capability to find the shortest path from the food source to the nest (destination) without any visual cues (Holldobler and Wilson, 1990). Ants can communicate with each other through chemical substance called pheromones in their immediate environment also, they are capable of adapting to change in the environment, for example finding a new shortest path once the old one is no longer feasible due to a new obstacle (Beckers, Deneubourg and Goss, 1992). The ants deposit some amount of pheromones on the ground while walking from nest to the food source or food source to the nest or vice versa this pheromone is the trivial factor for finding the best solution. An ant sense the pheromone deposited by the other (earlier) ants and tend to imitate the trail with stronger pheromone. So a shorter path having a higher amount of pheromone is probability and ants will tend to choose the shorter path. M. Dorigo applied this method to solve the classical TSP in 1992[1] and found that the algorithm has some advantages than others such as robustness and distributive nature. After that the ACO a meta-heuristic technique, widely applied to solve various combinatorial optimization problem such as Job-Shop Scheduling Problem, Vehicle Routing Problem, Network Routing Problem, Multiple Knapsack Problem etc. In this paper a new strategy is developed in which the large problem of the TSP is partitioned into the number of sub problems by creating the small number of cities in each group and then apply the ACS algorithm for the sub problems. The paper is organized as follows: section 2 describe the traveling salesman problem (TSP). In section 3 illustrates the ant colony system (ACS). In section 4 present the proposed algorithm for the TSP. In section 5 the proposed method is employed into several TSP problem and the result is compared with traditional ACO. The last section 6 makes the conclusion.

2 TRAVELING SALESMAN PROBLEM

The traveling salesman problem (TSP) is one of the well-known problem in discrete or combinational optimization. A traveling salesman problem is the problem in which a sales man have to visit all the cities exactly ones into a given number of city set and come back to the home city in such a way the total cost of visiting the cities should be minimum. TSP is an NP-Hard problem which means there is no particular solution for solving the TSP and it is very easy to describe but so difficult to solve. Graph theory defines the problem as finding the Hamiltonian cycle with the least weight for a given complete weight graph. It is widely spread in some industrial problem such as frequency assignment, machine scheduling, cellular manufacturing etc. can be formulated as a TSP. A general Travelling Salesman Problem can be represented by a complete weighted directed graph G = (V, E), where V = {1, 2, 3....n} is a set of vertices representing cities and E being the set of arcs fully connecting the nodes [8]. In addition, E is associated with the set D representing costs d, where dij is the metric of the distance between the cities i and j. dij can be defined within the Euclidean space and is given as follows:

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

The objective of the TSP is to find shortest closed tour in V such that each city is visited once and only once.

3 ANT COLONY SYSTEM

1. Background:

The Ant Colony System (ACS) builds on the Ant System, which is inspired from the foraging behavior of the real ant. Real ants are capable of finding the shortest path from foot source to the nest without any visual cues [10] this is called stigmergic communication between ants. The ants deposit some amount of chemical substance (called pheromone) on the ground while walking in searching of food source. This pheromone acting as memory preservation for ants in order to come up with the shortest path so this pheromone is useful for increasing the probability of other ants following the same path which are proportioned to the density of the pheromone. The more ants walk on a trail, the more pheromone is deposited on the ground and more ants follow the trail. Through this mechanism ants will find the shortest path. The figure 1 shows the behavior of the real ants. In figure 1A the real ants follow a path between nest and food source in a straight line. In fig 1B an obstacle appears on the path in this point ant choose whether to turn right or left with equal probability because there is no pheromone trail, at this point some ants will choose the upper path and some ants will choose the lower path. In fig 1C pheromone is deposited more quickly on the shorter path. After sometime all ants have chosen the shorter path as shown in fig 1D.

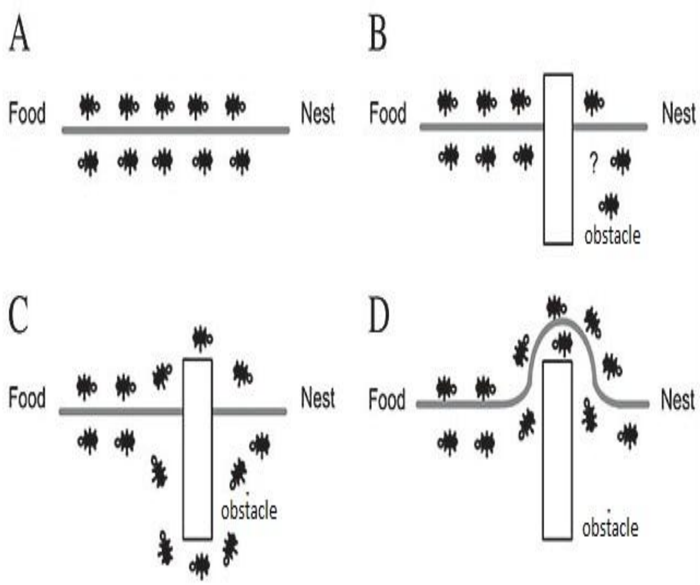

Fig 1.

A: Ants in a pheromone trail between nest and food.
B: an obstacle interrupts the trail.
C: Ants find two paths to go around the obstacle.
D: A new pheromone trail is formed along the shorter path

2. Mathematical Model of ACS:

The ant colony system is differ from the ant system in there aspects: first is the state transition rule (pseudo-random proportional rules) which provide the balance between exploration of new edges and exploitation of a priori. Second is the local pheromone updating rule is applied while ants construct a solution. And third is the global updating rule is applied only to edges which belong to the best ant tour. The ACS algorithm is reported in fig. 2, in the following we discuss the three aspects of ACS in brief.

2.1. ACS State Transition Rule:

An ant positioned on node r chooses the city s to move to by applying the given rule:

$$s = \begin{cases} \arg max_{u\epsilon\, j_k(r)}\{[\tau(r,u)]^{\alpha}.\ [\eta\,(r,u)]^{\beta}\}, \\ \qquad\qquad\qquad if\ (q\ \le q_0\ (exploitation) \\ S, \qquad otherwise\ (biased\ exploration) \end{cases} \quad \dots(1)$$

Where τ is the pheromone, $\eta = 1/d$ is the inverse of the distance d (r , s). $j_k(r)$ is the set of cities that remain to be visited by k ant positioned on city r. β is a parameter which determines the relative importance of pheromone density versus distance $(\beta > 0)$. q is a random number uniformly distributed in [0…..1], q_0 is a parameter ($0 \le q_0 \le 1$), the parameter q_0 determines the relative importance of exploitation versus exploration. S is a random variable selected according to the probability distribution given by:
Every time an ant in city r has to choose a city s to move to, it

$$P_k(r,s) = \begin{cases} \dfrac{[\tau\,(r,s)]^{\alpha}.\ [\eta\,(r,s)]^{\beta}}{\sum_{u\epsilon\, j_k(r)}[\tau\,(r,u)]^{\alpha}.\ [\eta\,(r,u)]^{\beta}}, if\ s\epsilon\ j_k(r) \\ 0, \qquad\qquad\qquad otherwise \end{cases} \quad \dots(2)$$

samples a random number q ($0 \le q \le 1$). If $q \le q_0$ then the best edge is chosen (exploitation) according to (1). Otherwise an edge is chosen according to (2) (biased exploration)

2.2. Local Pheromone Update Rule:

While construction a tour, an ant will modified the pheromone level on the visited edges using the local pheromone updating rule by:

$$\tau(r,s)\ \leftarrow (1-p).\tau(r,s) + p.\Delta\tau_0 \qquad \dots(3)$$

Where:

$$\Delta\tau_0 = \left({}^{1}/_{L_{mn}.n}\right)$$

Where Lmn is a tour length produced by nearest neighbor heuristic and n is the total no of cities.

2.3. Global Pheromone Update:

After all ants have constructed a tour, only the globally best ant which produces the shortest tour from the beginning of the trail will be allowed to do pheromone update using the global pheromone updating rule.

$$\tau(r,s)\ \leftarrow (1-p).\tau(r,s) + p.\Delta\tau(r,s) \qquad \dots(4)$$

Where:

$$\Delta\tau(r,s) = \begin{cases} \dfrac{1}{L_{gb}}, & if(r,s) \in Global\ best\ tour \\ 0, & otherwise \end{cases}$$

Where Lgb is the length of globally best tour from the beginning of the trail.

2.4 Candidate List Strategy

The candidate list is a strategy that is applied to the TSP to

```
Initialize
        Set the initial pheromone values for all edges,
populate the cost matrix with the distances between the
cities
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
        Untill all ants have built a complete solution
                A global pheromone updating rule is
                applied
Untill End_condition
```

Fig. 2

improve the performance of the ACS algorithm. It was proposed by Gambardella to accommodate searching procedure of ACS for larger data. In the ACS algorithm, when an ant choose the next city, the probability of its transfer from city i to city j needs to be computed, and then the city whose decision probability 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 of the candidate list. The algorithm for creating the candidate list is given below:

```
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)
```

4 PROPOSED APPROACH

To improve the performance of ACS algorithm, the proposed approach is based on the divide and conquer partition scheme. A 2-dimensional partitioning scheme was proposed by Karp which was similar to the construction of a k-dimensional tree data structure [11]. It involved geometrically partitioning of the cities and then applied local optimization for improvement. Alternative partitioning schemes were introduced by Reinelt [8] but the experimental result concluded that these approaches are not preferable to Karp's partitioning scheme. The ACS algorithm is better for problem of smaller size having less number of cities because of the random nature of the algorithm in which a large number of random decisions made on weighted choice are required to come together to construct an efficient solution as the size of the problem is increases. In the proposed approach the set of all the cities for a problem (S) is further sub divided into two disjoint sets S1 and S2 (In the proposed algorithm we use the nearby cities partition scheme to create the sub problems) and then proceed to find the solution of these two sub problems separately by applying the ACS algorithm using candidate list that generate a candidate tour. The candidate list applying only when the number of cities in the sub problems is very large or the number of cities is greater than 4. The overall solution or the original path is obtained by the conjunction of the solution given by these sub problems. Now focusing on reducing the length of the segments formed by the sub sets in the original problem in such a way that the nearby city of the two separate sub problems are joined at end point of the path and, after joining the end point of all the sub problems the original path can be obtained. The greedy edge exchange is used for optimizing the path by joining two sub problems. For joining the two sub problems we have chosen the closed city between the two sub problems and apply the path between these two cities and avoid the closed path of the sub problems for joining, repeat the same process for all sub problems to obtain the complete path. As the search space for these two sub problems get reduced because of the division of the cities from the original problem, the efficiency and accuracy of the ACS algorithm is much greater for these sub problems. The accuracy of the overall solution is upper bounded by the accuracy of the division of the cities for each sub sets, which is depend on the accuracy of the candidate tour. The proposed algorithm is combined with candidate list and the recursive approach that is applied to the sub problems to find the solution. Each ant builds a tour by repeatedly applying a stochastic greedy rule. While constructing the tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. Once ants have completed a tour they use their memory to evaluate the built solution and to retrace the same tour backward and increase the intensity of the pheromone trails $\tau(i,j)$. If j has not been visited previously, it can be selected with a probability that is proportional to the pheromone associated with edge (i,j). In the proposed system all the ants deposit pheromone. Once all ants have terminated their tour, the amount of pheromone on edges is modified again. Ants are guided in building their tour, by both heuristic information and by pheromone information. An edge with a large amount of pheromone is a very desirable choice. The proposed algorithm is:.

5 EXPERIMENTAL RESULT

The algorithm has been implemented in .NET using C# and all problem instances studied as test cases in this paper are TSPs taken from TSPLIB [24]. In this study, we compared our proposed algorithm results with those of the ACS algorithm in the aspects of algorithm convergence and experiment results.

Step 1: Partitioning the problem into N number of sub problems using the nearby city strategy, with the help of cost matrix.
Step 2: Initialization of parameters for each sub problems
Step 3: for sub problems 1 to N
  Loop /* this level loop is called iteration*/
  Each ants is positioned on a starting node according to distribution strategy.
  For k=1 to m /* this level called step*/
    Repeat:
      Compute Candidate List /* for the large group size or group size>4.*/
      Select node j to be visited next according to eq. (3)
    Until ant k has completed a tour
  End for
  Apply the local search to improve tour.
  A local updating rule is applied as eq (5)
  Until end_condition
  End for
Step 4: for each sub problems
  Find the closed city between two sub problems using cost matrix.
  Apply edge exchange for the local optimization of the tour.
  Apply Global updating rule as eq. (4).

For each test case of TSP 10 trails were conducted. The performance of the algorithm is based on the correct tuning of the parameters. We set the parameters for the ACS as follows: $\alpha = 1$, β is the dynamically value for the proposed algorithm and β =2 for the TSP, $\rho = 0.75$, q_0 = .95 and the number of ants is 30. In the following table 1 we have recorded the effect of the division of cities into sub problems for the Eil51. The result is obtain by applying the different group size to the problem. The given table shows that the result obtain by the group size of 5,12 and 18 are equal but the best result is obtain by the group size of 12 because it has minimum number of pheromone update count..

***TABLE 1** Result Obtained for TSP Eil51*			
S. No.	**Group Size**	**Result Obtained**	**Pheromone update count**
1	3	443.06	96
2	**5**	**433.59**	**88**
3	7	439.46	64
4	10	455.13	76
5	**12**	**433.59**	**65**
6	15	439.46	68
7	**18**	**433.59**	**78**
8	20	443.02	72
9	22	455.26	73
10	25	441.87	74

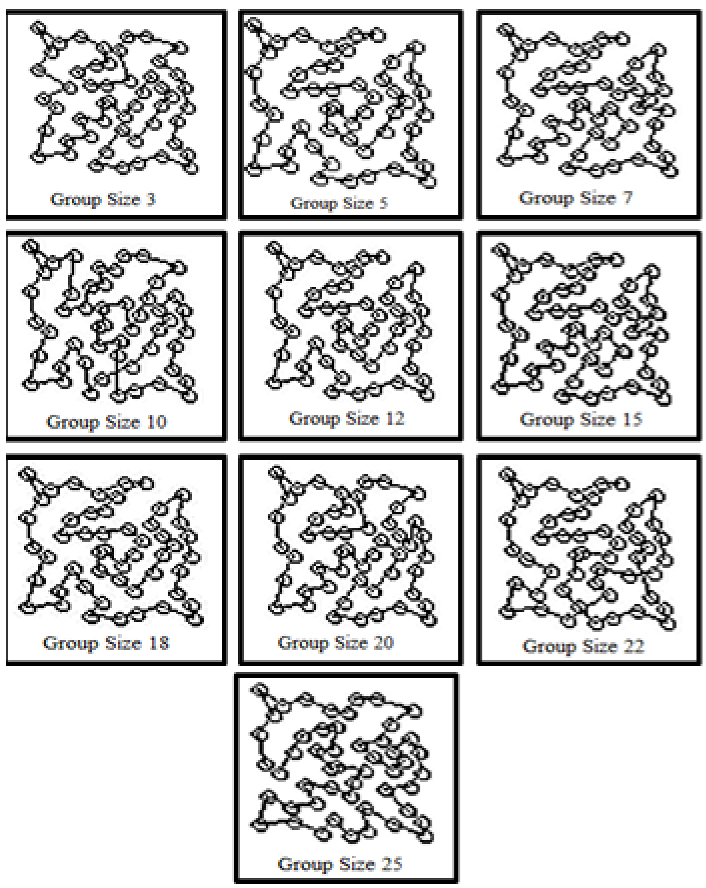

***Fig 2.** Path obtained for Eil51*

***TABLE 2** Simulation of the results*

Problem Domain	Number of Nodes	Known Optimal Value	ACS Value	Proposed Algorithm				
				Group Size	Best Result Obtained	Worst Result Obtained	Pheromone Update Count	Average
Eil51	51	426	456	12	433.59	459.41	65	442.21
Berlin52	52	7544	7549	7	7544.37	8089.46	105	7784.39
PR76	76	108159	11718 1	3	110932.70	113143.67	116	111960.12
ST70	70	675	706	7	688.78	781.19	165	718.59

The experiment result showed in the simulation table 2. The provided table summarize the results of the proposed algorithm on a few test cases and compares it with the results obtained by the ACS component of the algorithm by itself. The first column of the table is the problem domain for the different TSP, second column having the number of nodes (cities) in the problem, in the third and fourth column we reported the known optimal solution and the result obtain by running the ACS algorithm in references [15], the fifth column is the proposed algorithm and its having five parts, in the first part we reported the best group size of the sub problem to obtain the best result, in the second part we report the best result obtain by the respective algorithm, the third column reported the worst result obtained by the respective algorithm, fourth part shows the pheromone update count, and the last column of the table show the average of the results obtain by performing the proposed algorithm for the different group size up to 10 trails.as shown in table 1 for the Eil51.

6 Conclusion

This paper presents an efficient approach for solving traveling salesman problem based on the divide and conquers strategy (in which cities are divided into the number of sub city) in which the ant colony algorithm using the candidate list is applied. From our experimental result the proposed algorithm is more effective than the conventional ant colony algorithm to find the better solution. Future work is to apply the proposed algorithm to other combinational problems and check the efficiency by comparing this method with other proposed method.

References

[1] Dorigo M., Maniezzo V. & Colorni A., “The Ant System: Optimization by a colony of cooperating agents,” IEEE Trans. Systems, Man, and Cybernetics- Vol.26,N 1, pp.1-13, 1996.

[2] Dorigo M, Gambardella L M. Ant Colony System: a Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1997, 1(1) 53-66.

[3] A. Colorni, M. Dorigo, V. Maniezzo, "Distributed optimization by ant colonies".Proceedings of European Conference on Artificial Life, Paris, France, pp. 134-142, 1991

[4] C.P. Ravikumar. "Solving Large-scale Travelling salesperson Problems on Parallel Machines". Microprocessors and Microsystems 16(3), pp. 149-158, 1992.

[5] Y.H. Song, K.Y. Lee, J.G. Vlachogiannis, Y.H. Lu and I.K. Yu “Ant colony search algorithms in power system optimization,” In “Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems,” Eds. K.Y. Lee and M.A. El-Sharkawi, IEEE Press Book

[6] Yingzi Wei Yulan Hu, Kanfeng Gu “Parallel Search Strategies for TSPs using a Greedy Genetic Algorithm” Third International Conference on Natural Computation IEEE 2007.

[7] Dorigo M. & Di Caro G., “AntNet: Distributed tigmergetic Control for Communication Networks,” Journal ofArtificial Intelligence Research, Number 9, pp 317-365, 1999.

[8] G. Reinelt. The Traveling Salesman: Computational Solutions for TSP Applications, volume 840 of LNCS. Springer Verlag, 1994.

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

[10] B. Hölldobler and E. O. Wilson, The Ants. Berlin: Springer-Verlag, 1990.

[11] R.M. Karp, “Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman in the Plane”, Math. Oper. Res. 2 (1977), 209-224.

[12] Stutzle T, Hoos H H. Max-Min Ant System and Local Search for the Traveling Salesman Problem II Proceedings of IEEE International Conference on Evolutionary Computation. New York,USA: IEEE Press, 1997 309-314.

[13] Christian Nilsson, “Heuristics for the Traveling Salesman Problem”.

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

[15] MA Ying, TIAN wei-jian and FAN yang-yu, “Improved quantum ant colony algorithm for solving TSP

problem",2014 IEEE workshop on electronics, computer and application

[16] Ramlakhan Singh Jadon, Unmukh Datta, "Modified Ant Colony Optimization Algorithm with Uniform Mutation using Self-Adaptive Approach for Travelling Salesman Problem" 4th ICCCNT IEEE 2013.

[17] Guiqing Liu, Juxia Xiong, "Ant Colony Algorithm Based on Dynamci Adaptive pheromone Updating and its Simulation." Sixth International Symposium on ComputationalIntelligence and Design. 2013 IEEE.

[18] Dong-Hyun Lee and Ju- Jang Lee "Global path planning using improved ant colony optimization algorithm through bilateral cooperative exploration " IEEE 2011

[19] Cai Shukai and Jiang Yanbin "A Brief review of Ant Algorithms" The 1st International Conference on Information Science and Engineering IEEE 2009.

[20] Xiao-Min Hu, Jun Zhang, Henry Shu-Hung Chung, Yun Li "SamACO: Variable Sampling Ant Colony Optimization Algorithm for Continuous Optimization" ieee transactions on systems, man, and cybernetics—part b: cybernetics, vol. 40, no. 6, december 2010.

[21] Sorin Ilie and Costin Badica "Effectiveness of Solving Traveling Salesman Problem Using Ant Colony Optimization on Distributed Multi-Agent Middleware" Proceedings of the International Multiconference on Computer Science and Information Technology 2010 IEEE.

[22] Helmi Md Rais, Zulaiha Ali Othman, Abdul Razak Hamdan "Reducing Iteration Using Candidate List" 2008 IEEE.

[23] K. Sheibani "The Fuzzy Greedy Search in Combinatorial Optimization with Specific Reference to the Travelling Salesman Problem" 2010 IEEE

[24] Reinelt, G.: TSP-Library (TSPLIB) (2008). http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms
https://en.wikipedia.org/wiki/Travelling_salesman_problem

Citation: Raghavendra BV (2015) Solving Traveling Salesmen Problem using Ant Colony Optimization Algorithm. J Appl Computat Math 4:260. doi:10.4172/2168-9679.1000260. Copyright: Â© 2015 Raghavendra BV.Â Figure 1: Flow Chart for ACO. The traveling salesman problem (TSP) is the problem of finding a shortest closed tour which visits all the cities in a given set. In a symmetric TSP the distance between two cities is the same regardless of the direction of travel whereas in the asymmetric TSP the distance is different with regards to the direction of travel [4]. This paper restricts attention to symmetric TSPs in which cities are on a plane and a path (edge) exists between each pair of cities. 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.Â 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. I. The travelling salesman problem (TSP) is one of the most intensively studied problems in optimization. Loosely speaking, given a set of cities on a map, the problem consists of finding a tour that goes through each city exactly once and ends in the same city it started with.Â Thus we opt for the second goal. We use an algorithm termed the Ant Colony Optimization algorithm that simulates the way ants find the shortest route to a food source. There already exist sequential versions of this algorithm.Â In computer science and operations research, the ant colony optimization (ACO) algorithm is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.