Compute Travelling Salesman Problem Using Ant Colony Optimization

*Megha Aggarwal, **Dr. Saroj
*M.Tech CSE(Student), **Associate Professor
Guru Jambheshwar University of Science and Technology, Hisar
megha.06.aggarwal@gmail.com, ratnoo.saroj@gmail.com

Abstract: Combinatorial optimization problem are often easy to state but very difficult to solve. ACO is an intelligent metaheuristic optimization algorithm with strong robustness & the ability to find the optimal solution which gives it prominence to optimization problems. TSP is one of most intensively studied NP-hard problem in combinatorial optimization environment. The basic high-level view of an ACO algorithm for TSP are studied in this paper. The algorithm is realized under the MATLAB tool & applied to solve the TSP.
Keywords: Ant colony optimization(ACO); Travelling Salesman Problem(TSP); Pheromone; Combinatorial optimization(CO); TSPLIB.

I. INTRODUCTION

Ant colony optimization (ACO) was introduced by M. Dorigo and colleagues as a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization (CO) problems. ACO belongs to the class of metaheuristics [1], which are approximate algorithms used to obtain good enough solutions to hard CO problems in a reasonable amount of computation time. The inspiring source of ACO is the foraging behavior of real ants. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the ant deposits a chemical pheromone trail on the ground. The quantity of pheromone deposited will guide other ants to the food source. As it has been shown in [2], indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources. This characteristic of real ant colonies is exploited in artificial ant colonies in order to solve CO problems.

The ACO algorithm, has been applied to the Traveling Salesman Problem (TSP) which is the problem of finding a shortest closed tour which visits all the cities in a given set [3].

In this paper we give an overview of ACO algorithm for the TSP. We first introduce, in Section II, the ACO. Section III briefly discusses TSP. In Section IV we outline how ACO algorithm can be applied to TSP. At last, Section V gives the experimental results of ACO for TSP under the MATLAB tool.

II. ANT COLONY OPTIMIZATION

ACO [4] is a class of algorithms, whose first member, called Ant System, was initially proposed by Colorni, Dorigo and Maniezzo [5][6][7]. The original idea comes from observing the exploitation of food resources among ants, in which ants' individually limited cognitive abilities have collectively been able to find the shortest path between a food source and the nest. An ant (called "blitz") runs more or less at random around the colony. If it discovers a food source, it returns more or less directly to the nest, leaving in its path a trail of pheromone; these pheromones are attractive, nearby ants will be inclined to follow, more or less directly, the track. Returning to the colony, these ants will strengthen the route. If there are two routes to reach the same food source then, in a given amount of time, the shorter one will be travelled by more ants than the long route. The short route will be increasingly enhanced, and therefore become more attractive. The long route will eventually disappear because pheromones are volatile. Eventually, all the ants have determined and therefore "chosen" the shortest route [4].

III. TRAVELLING SALESMAN PROBLEM

Intuitively, the TSP is the problem of a salesman who, starting from his hometown wants to find a shorter tour that takes him through a given set of customer cities & then back home, visiting each customer city exactly once. The TSP can be represented by a complete weighted graph G=(N,Z) with N being the set of nodes representing the cities, & Z being the set of arcs [8][9]. Each arc $(i,j) \in Z$ is assigned a value (length) d_{ij} , which is the distance between cities i & j, with $(i, j) \in N$. The goal of the TSP is to find a minimum length Hamiltonian circuit of the graph, where a Hamiltonian circuit is a closed path visiting each of the $n=|N|$ nodes of the G exactly once [10]. The main reasons for the choice of TSP for ACO are [11]:-> TSP is an important NP-hard optimization problem that arises in several applications. It is a problem to which ACO algorithms are easily applied.

-> It is easily understandable, so that algorithm behavior is not obscured by too many technicalities .

->It a standard test bed for new algorithmic ideas- a good performance on the TSP is often taken as a proof of their usefulness.

->TSP is a typical combinatorial optimization problem. It is often used to validate a certain algorithm, making the comparison with other algorithms an easy work.

All the TSP instances we use are taken from the TSPLIB Benchmark library [12] which contains a large collection of instances; these have been used in many other studies or some from practical applications of the TSP. TSPLIB is accessible on the WWW at the address http://www.iwr.uniheidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html.

IV. ANT COLONY ALGORITHM FOR TSP

In ACO algorithms ants are simple agents which, in the TSP case, construct tours by moving from city to city on the problem graph [8]. It chooses the city to move to using a probabilistic function both of trail accumulated on edges and of a heuristic value, which was chosen here to be a function of the edges length. Artificial ants probabilistically prefer cities that are connected by edges with a lot of pheromone trail and which are close-by. Initially, m artificial ants are placed on randomly selected cities. When all the ants have completed a tour the ant that made the shortest tour modifies the edges belonging to its tour –termed global trail updating– by adding an amount of pheromone trail that is inversely proportional to the tour length [3].

Tour Construction: Initially, ants are put on randomly chosen cities. At each construction step, ant k applies a probabilistic action choice rule, called random proportional rule, to decide which city to visit next. In particular, the probability with which ant k, currently at city i, chooses to go to city j is

$$\mathbf{p_{ij}^{k}} = \frac{(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})}{\Sigma(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})} \text{ , } \mathbf{if\ j} \in \mathbf{N_i^k}$$

where $\eta_{ij} = 1/ d_{ij}$ is a heuristic value that is available a priori, α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information, and N_i^k is the feasible neighborhood of ant k when being at city i, i.e.,the set of cities that ant k has not visited yet(the probability of choosing a city outside N_i^k is 0). Each ant maintains a memory M^k which contains the cities already visited, in the order they are visited. This memory is used to define the feasible neighborhood N_i^k in the construction rule given above. In addition, the memory M^k allows ant k both to compute the length of the tour T^k it generated and to retrace the path to deposit pheromone.

Pheromone Update: After all the ants have constructed their tours, the pheromone trails are updated which is done by first lowering the pheromone value on all arcs by a constant factor, and then adding pheromones on the arcs they have crossed in their tours. Pheromone evaporation is given by

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \ \square(i, j) \in L$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate. The parameter ρ is used to avoid unlimited accumulation of the pheromone trails. After evaporation, all ants deposit pheromone on the arcs they have crossed in their tour:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}, \ \square(i, j) \ \square \ L$$

where $\Delta\tau_{ij}^{k}$ is the amount of pheromone ant k deposits on the arcs it has visited. It is defined as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{C^k}, \text{if arc}(i, j) \text{belongs to } T^k; \\ 0, \text{otherwise}; \end{cases}$$

where C^k, the length of the tour T^k built by the K-th ant, is computed as the sum of the lengths of the arcs belonging to T^k. By the immediate above equation, the better an ant's tour is, the more pheromone the arcs belonging to this tour receive [8]. The high-level view of an ACO algorithm for the TSP is given below (Fig.1):

Fig.1:High-level view of an ACO algorithm for the TSP

```
procedure ACOforTSP
            InitializeData

  While (not terminate) do
        ConstructSolutions
        LocalSearch
        UpdateStatistics
        UpdatePheromoneTrails

end-while
end-procedure
```

In general, arcs that are used by many ants and which are part of short tours, receive more pheromone and are therefore more likely to be chosen by ants in future iterations of the algorithm.

V. RESULTS & COMPARISONS

We executed ACO on a well-known problem instance of classical city dataset from TSPLIB, which is att48 (48 capitals of the US), under the MATLAB tool. We set the parameters according to many experiments, which include α=1,β=2, ρ=0.65 & Q=10. The maximum no. of iterations is 500. Running the program & calculating, we can see the result path (Shortest Path) in Fig.2 while Fig.3, show the iterative best cost over the 500 iterations.

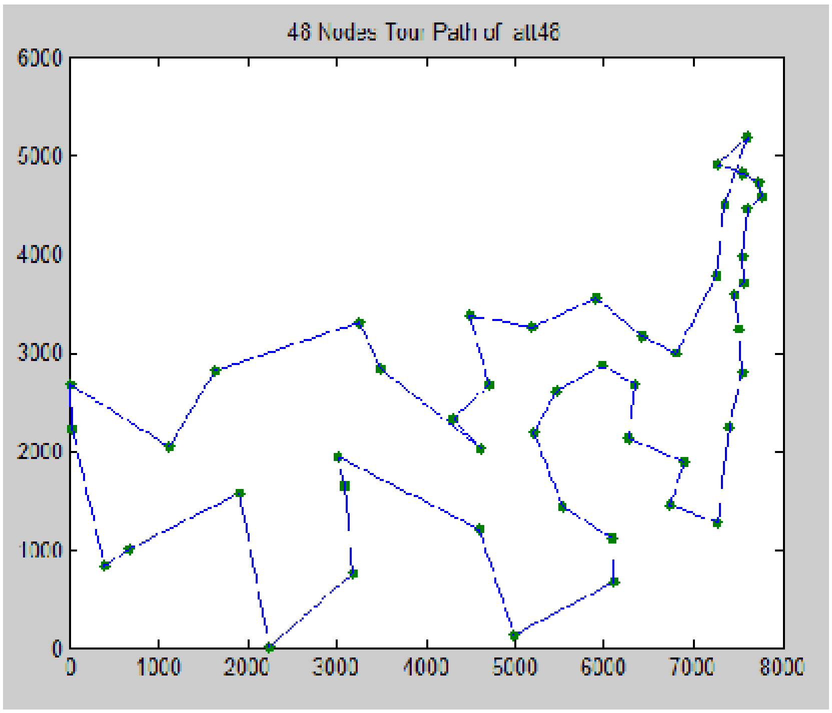

Fig.2: Shortest Path

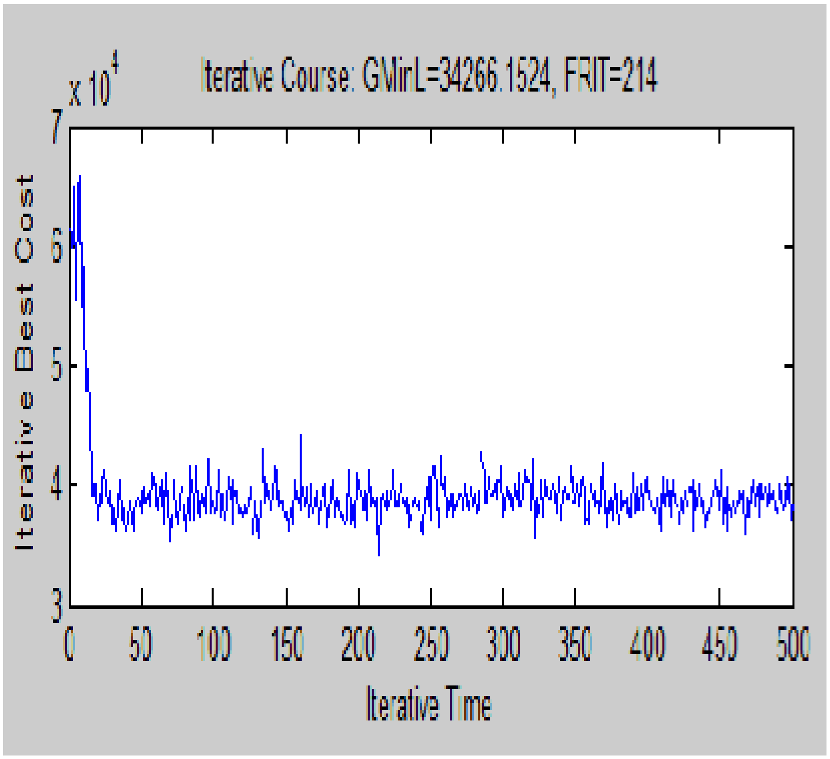

Fig.3: Iterative Best Cost

The path length computed by ACO for att48 dataset of TSPLIB is 34266.15 which is very close to its optimal solution. We also check how the efficiency of ACO varies with the no. of ants and with pheromone importance parameter. All the values obtained from experimental results are rounded off to nearest integer value for better understandability and readability.

TABLE I. THE AFFECTION OF THE NUMBER OF ANTS TO THE SOLUTION

CityNum	48	48	48	48	48
AntNum	10	15	20	40	48
Tour length	35307	35238	34985	34745	34266
Time(sec)	77	128	153	323	350

α=1,β=2,ρ=0.65,Q=10,Maxiter=500

TABLE I investigates the role of the number of ants m on the final performance of ACO. We ran ACO using parameter settings of m {10,15,20,40,48} leaving all other choices the same. The above table clearly shows that the most optimal path is obtained when the no. of ants is equal to the number of cities. Moreover, as the no. of ants increase the time to find the shortest path will also increase.

TABLE II. THE AFFECTION OF PHEROMONE PARAMETER

CityNum	48	48	48	48	48
α	0.1	0.3	0.5	0.8	1
Tour Length	45711	42165	38788	35523	34266

$m=48, \beta=2, \rho=0.65, Q=10, Maxiter=500$

TABLE II shows that the best optimal path is obtained when α is 1 or very close to 1. Moreover, if α = 0, the closest cities are more likely to be selected which corresponds to a classic stochastic greedy algorithm. If α > 1 it leads to the rapid emergence of a stagnation situation, i.e., a situation in which all the ants follow the same path and construct the same tour, which, in general, is strongly suboptimal.

VI. CONCLUSION

This paper is success to solve the Travelling Salesman Problem by ACO under the MATLAB environment. The experimental results show the good optimization capability of ACO. The Performance of ACO for TSP has been analysed under various parameters which shows that most optimal paths are obtained when the number of ants is equal to the no. of cities as well as the value of α should be close to 1. In addition, we also find that the ACO has a problem of stagnation, how to combat this problem is worthful to study.

REFERENCES

[1] Dorigo and C. Blum, "Ant colony optimization theory: A survey," Theoretical Computer Science, vol. 344, no. 2–3, pp. 243–278, 2005.

[2] Blum, "Ant colony optimization: Introduction and recent trends," Physics of Life Reviews, vol. 2, no. 4, pp. 353–373, 2005.

[3] M. Dorigo and L. M. Gambardella, "Ant Colonies for the Traveling Salesman Problem," 1997.

[4] Wikipedia contributors, "Ant colony optimization algorithms," Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 16-Jun-2012.

[5] V. Maniezzo, L. M. Gambardella, and F. De Luigi, "Ant Colony Optimization," OPTIMIZATION TECHNIQUES IN ENGINEERING. SPRINGER-VERLAG, vol. 141, pp. 101–117, 2004.

[6] A. Colorni, M. Dorigo, V. Maniezzo, and others, "Distributed optimization by ant colonies," in Proceedings of the first European conference on artificial life, 1991, vol. 142, pp. 134–142.

[7] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: An autocatalytic optimizing process," TR91-016, Politecnico di Milano, 1991.

[8] T. Stützle and M. Dorigo, "ACO algorithms for the traveling salesman problem," Evolutionary Algorithms in Engineering and Computer Science, pp. 163–183, 1999.

[9] Kangshun Li, Lanlan Kang, Wensheng Zhang, and Bing Li, "Comparative Analysis of Genetic Algorithm and Ant Colony Algorithm on Solving Traveling Salesman Problem," in IEEE International Workshop on Semantic Computing and Systems, 2008. WSCS '08, 2008, pp. 72–75.

[10]Wikipedia contributors, "Travelling salesman problem," Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 15-Jun-2012.

[11] Lanlan Kang and Wenliang Cao, "An Improved Genetic & Ant Colony Optimization Algorithm for Travelling Salesman Problem," in 2010 International Symposium on Information Science and Engineering (ISISE), 2010, pp. 498–502.

[12] G. Reinelt, "TSPLIB—A Traveling Salesman Problem Library," ORSA Journal on Computing, vol. 3, no. 4, pp. 376–384, Sep. 1991.