A SURVEY ON ANT COLONY OPTIMIZATION

Jaskiran kaur
Computer Science And Engineering Department
DAV Institute of Engineering and Technology
Jalandhar, India
jaskiranboparai@gmail.com

Inderpal Singh
Computer Science And Engineering Department
DAV Institute of Engineering and Technology
Jalandhar, India
Er.inderpal13@gmail.com

Abstract: This paper deals with Ant Colony Optimization, a heuristic algorithm with strong robustness and the ability of finding the optimal solution which has been applied to a number of combinatorial optimization (CO) problems, of which the most important one is the traveling salesman problem (TSP). Ants of the colony have the ability to generate shorter feasible tours through information, which is accumulated, in the form of a pheromone trail deposited on the TSP graph's edges. For solving the TSP problem, ACO is one of the high performance computing methods but still has some drawbacks, which include stagnation behavior, computational time, which is longer, and premature convergence problem.

Keywords: Ant Colony Optimization, Traveling salesman problem (TSP), Pheromone, Combinatorial optimization (CO)

I. INTRODUCTION

Ant colony optimization (ACO) was introduced by the Italian scholar [1], M. Dorigo and colleagues. It is a novel meta-heuristic technique [2] that has been successfully applied in solving various problems in combinatorial optimization. To establish the shortest path from food sources to nests [2], ACO algorithm helps in modelling the behaviour of real ants. The ants deposit pheromone trail while walking and all other ants prefer to follow a path where the amount of pheromone is rich [2,3,5]. When an ant searches a food source, it carries it back to the nest and starts depositing the chemical. Other ants will tend to select a shorter path between food source and their nest , where there is higher quantity of pheromone. This ant foraging behaviour can solve CO problems [1].

The ACO algorithm has been widely applied to various CO problems [2,4] such as Traveling Salesman Problem (TSP), Job-shop Scheduling Problem (JSP), Vehicle Routing Problem (VRP), Quadratic Assignment Problem (QAP), Weapon-Target Assignment problems (WTA), etc. and enjoys a rapidly growing popularity [4].

The Traveling Salesman Problem(TSP) is a typical combinatorial optimization problem [1] of finding a shortest closed tour [3], visiting all the cities in a given set. Consider given n cities [1], a salesman starts from a city, then visits all other cities, but, each and every city is visited only once, at last, the salesman goes back to the initial city, his path is a closed tour , and length is given by the addition of the lengths of all the arcs , it is composed of. Since the total number of paths grows with the number of cities , it is difficult to find the shortest path. This paper focuses on Heuristic algorithms [1] works well in solving this problem.

II. TRAVELING SALESMAN PROBLEM

Traveling Salesman Problem (TSP) is a well known and a typical non-deterministic polynomial complete problem [1] which is difficult to solve and easy to describe. It asks for the shortest path of minimal total cost [2] visiting each given city only once. The main goal of TSP is to find the Hamiltonian cycle [2,3] with the least weight for a graph where a Hamiltonian cycle is a closed path which visits each of the cities (nodes) of the graph exactly once. TSP is a problem to which ACO algorithms can be easily applied. It is easy to understand and is often used to validate certain algorithm by making an easy comparison with other algorithms [3].

The Traveling Salesman Problem can be represented by a complete weighted graph $G=(N,E)$, where N denotes the set of nodes representing cities, and E denotes the set of edges or arcs connecting all the cities [2,3]. Each arc $(i,j) \in E$ is assigned a cost(value or length) dij, that is the difference between cities i and j with $(i, j) \in N$ [2,3,4].

To select the path that has minimum total cost for all feasible permutations [4] of N cities is a direct method for solving problem. The permutations can be very large in number for even 40 cities. Each and every tour is represented in 2n different ways for symmetrical TSP. Since the possible ways for the permutation of n numbers is n! , the size of the search space is *|S|=n!/(2n)=(n-1)!/2*.

III. ANT COLONY ALGORITHM FOR TSP

A. ANT COLONY OPTIMIZATION

The Ant Colony Optimization techniques have emerged as a novel meta-heuristic [4] for combinatorial optimization problems. It is designed to determine the capability of ants in finding the shortest paths to food. Individual ants posses few capabilities but their work as a colony is capable of complex behaviour.

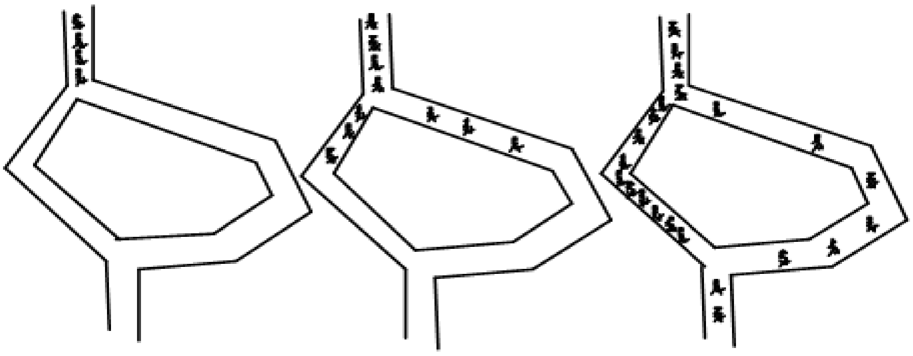

Fig. 1 . finding shortest path

Ants in the real world can communicate with each other indirectly by pheromone trial [2,3,4] without the use of visual clues and are able to find the shortest path between food and their nests. While walking, ants deposit the pheromone, and all other ants get attracted towards it based on the probability which is proportional to the amount of pheromone. More the ants walk on a trail, more will be the pheromone deposition on it and additional ants can follow the route. Through this technique, ants will be able to find the shortest path. Artificial ants copy the way of real ants, how they forage the food, but can solve much difficult problems that the real ants can[4]. An algorithm with this type of approach is called Ant Colony Optimization. Figure 1 descibes how shortest path is searched by the ants.

B. ANT SYSTEM

The main characteristic of the Ant System is that the pheromone values are updated by all the ants that have constructed a solution. Initially each city is assigned an ant [2,4] and each ant selects the next city depending on a probability .The probability that ant K will move from city i to city j is in accordance with the probabilistic decision rule and is given by :-

$$P_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{u \in j_k(i)} \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)} & \text{if } j \in j_k(\text{i}) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where $\tau_{ij} = amount\ of\ pheremone\ ant\ K\ deposits\ o \ldots city\ i\ and\ j.$

$$\eta_{ij} = heuristic\ information \eta_{ij} = \frac{1}{d_{ij}}\ where\ d_{ij} = distance\ between\ cities\ i\ and\ j$$

$j_k(i)$=set of cities that ant K has not yet visited when it is at city i

$$\alpha = \ Pheremone\ \ affects\ the\ path\ selection\ and\ this\ can\ be\ expressed\ by\ \alpha$$

$$\beta = Path\ length\ affects\ the\ path\ selection\ and\ this\ can\ be\ expressed\ by\ \beta$$

In each iteration, the pheromone value is updated by all the m ants that have built a solution in that iteration. When an ant traverses the complete tour, the pheromone concentration [2] on the path needs to be updated in accordance with below rule:-

$$\tau_{ij}(t+1) = \ (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}\,(t) \quad (2)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}\,(t) \quad (3)$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k} & , if\ Kth\ ant\ uses\ edge(i,j) in\ its\ tour \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

$\ldots mone\ volatization\ coefficient$

$$1-\rho = pheromone\ residual\ coefficient$$

$\Delta\tau_{ij}^{k}(t)$ =amount of pheromone that ant k leaved in the path from city i to city j

L_k =total length of the tour that each ant went through.

m=number of ants

Q=constant value (amount of pheromone that an can release in one iteration)

C. ANT COLONY SYSTEM ALGORITHM

The ACS [2,4] introduces a local pheromone update rule in addition to the global pheromone update. At the end of each construction step each ant performs the local pheromone update. The update is performed only on the edge last traversed.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\tau_o \quad (5)$$

$\rho\epsilon(0,1)$ *is the pheromone decay coefficient*

τ_o *is the intial value of pheromone*

Reducing the pheromone concentration on the edges allows the ants to choose different paths and hence produce different solutions. This is the main goal of performing the local pheromone update.

The global pheromone update is performed by one ant at the end of each iteration, which can be iteration best or best so far (edges which are visited by the best ants). The update is performed in accordance with the following rule:-

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \quad (6)$$

$$\Delta\tau_{ij}(t) = \begin{cases} 1/L_{gb} & , if\ (i,j) belongs\ to\ global\ best\ tour \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

$\frac{1}{L_{gb}}$ *is the length of the* globally best tour found.

Another difference between Ant System and Ant Colony System is the decision rule used by ants to move from city i to city j called Pseudo Random Proportional rule. The probability of an ant to move from one city to another depends on a random variable q uniformly distributed over [0,1] and a predefined parameter q_o.

$$j = \begin{cases} argmax_{u\epsilon allowed_k}(i)\left\{[\tau_{iu}^{\alpha}][\eta_{iu}^{\beta}]\right\} & if\ q < q_o \\ i & \text{Otherwise} \end{cases} \quad (8)$$

j is a random variable determined in accordance Otherwise on 1.

D. MAX-MIN ANT SYSTEM

This Algorithm [6] has proven to be an improvement over the original Ant System. In each iteration the pheromone updates are performed only by the best ant. The pheromone update rule in MAX-MIN ANT SYSTEM is as below:

$$\tau_{ij} = \left[(1-\rho).\tau_{ij} + \Delta\,\tau_{ij}^{best}\right]_{\tau_{min}}^{\tau_{max}} \quad (9)$$

Where

τ_{min} *and* τ_{max} *are the lower bound and upper bound of the pheromone values* .

$[y]_b^a$is defined as:

$$[Y]_b^a = \begin{cases} a & if y > a \\ b & if\ y < b \\ y & otherwise \end{cases} \quad (10)$$

$$\text{and } \Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \text{if (i,j) belongs to the best tour} \\ 0 & \text{Otherwise} \end{cases} \quad (11)$$

L_{best} *is the tour length of* Otherwise this can be best tour in the current iteration or the best solution found so far.

IV. COMPARISON OF RESULTS

(a) ACO was executed on a problem of city dataset [1] which is eil51. The parameters were set according to experiments, which include α=1, β=5, ρ=0.3 & Q=100. The maximum no. of iterations is 500. Running the program & calculating, the result path (Shortest Path) in Fig.2 could be seen. In Fig.3, the length of the shortest path is shown by the blue line and the average length of all the paths in every iteration process by the green line[1].

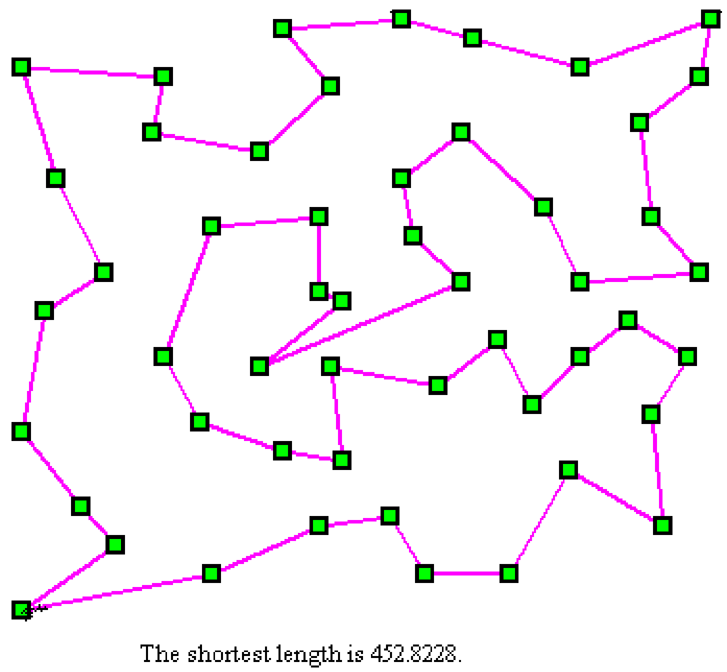

Fig.1 The shortest path

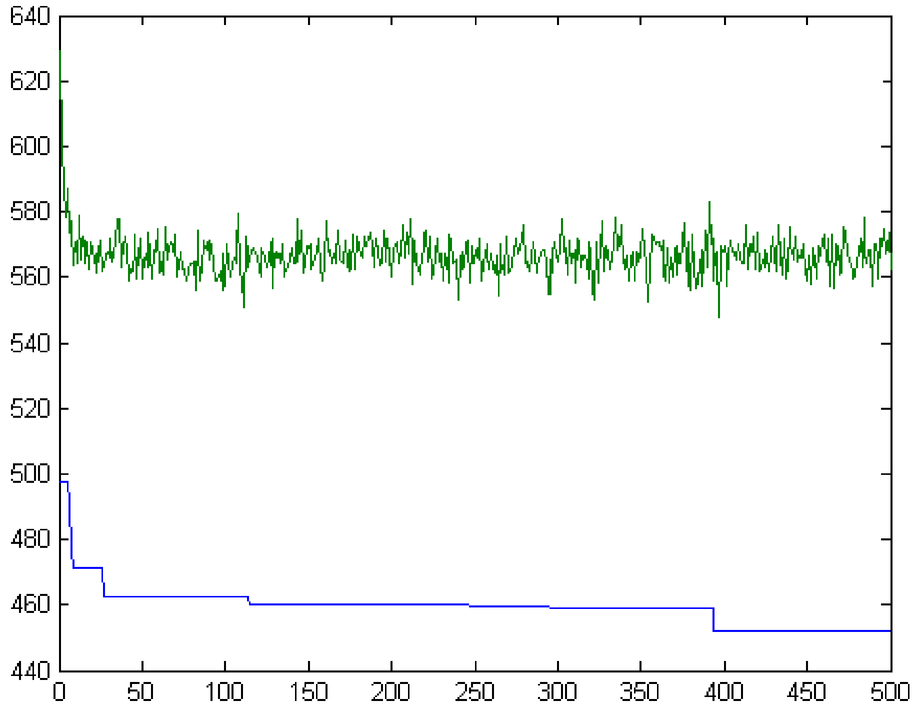

Fig.2 The iteration show

Now, the relationship between the number of ants and cities is shown in Table 1. When the number of ants is one or two times the number of cities, shorter path is found. Then the optimal solution changes with α shown in table 2.

Table 1. THE AFFECTION OF THE NUMBER OF ANTS TO THE SOLUTION

CityNum	51	51	51	51	51	51	51
AntNum	10	20	50	55	100	200	300
PathLength	481	468	466	465	460	457	457

$\alpha=1;\beta=5;\rho=0.95$; Itermax=500; Q=100;

Table 2. THE AFFECTION OF THE PHEROMONE IMPORTANCE PARAMETER

A	0.1	0.2	0.3	0.5	0.7	0.8	0.9	1
PathLength	481	479	467	466	461	461	461	460

β=5;ρ=0.95; Itermax=500; Q=100;AntNum=100;

(b) Executed ACO on a problem of city dataset [3] which is att48. The parameters were set , which include α=1, β=2, ρ=0.65 & Q=10. The maximum no. of iterations is 500. Running the program & calculating, the result path (Shortest Path) in Fig.4 could be seen. Fig.5 shows the iterative best cost over 500 iterations.

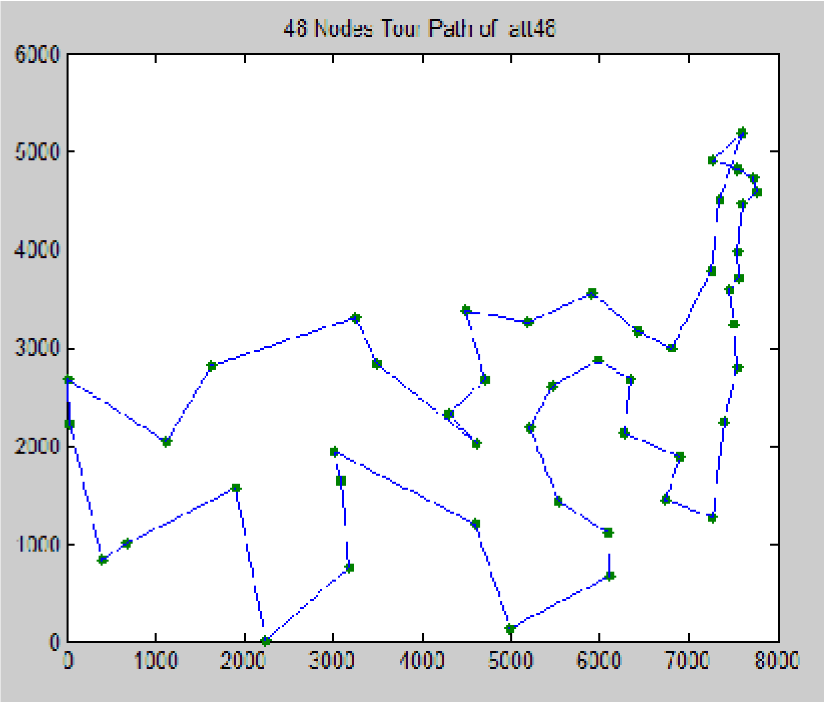

Fig.4: the shortest path

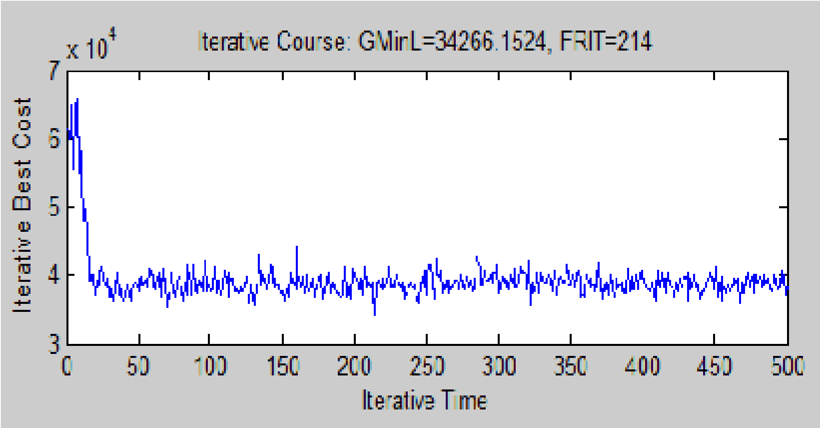

Fig.5: the Iterative best cost

Also, checked how ACO efficiency varies with the number of ants and pheromone importance parameter.

Table 3: THE AFFECTION OF THE NUMBER OF ANTS TO THE SOLUTION

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

α=1,β=2,ρ=0.65,Q=10,Maxiter=500

Table 4: THE AFFECTION OF THE PHEROMONE IMPORTANCE PARAMETER

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

m=48,β=2,ρ=0.65,Q=10,Maxiter=500

Table 4 demonstrates that when α = 1 or very close to 1, the best optimal route is obtained. If α=0, the closest cities are selected.

V. CONCLUSION

This paper focuses on solving traveling salesman problem based on ant colony algorithm. It analyzes different experimental data and different values of parameters, which are useful for the future study of the algorithm. It shows comparison of results based on two problems of cities from eil51 and att48 files.

REFERENCES

[1] ZHU Ju-fang and LI Qing-yuan "Solving the Traveling Salesman Problem by the Program of Ant Colony Algorithm,",IEEE,2009
[2] Zar Chi Su Su Hlaing and May Aye Khine "An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem," *International Conference on Information Communication and Management, IPCSIT,* vol.16,2011
[3] Megha Aggarwal and Dr. Saroj "Compute Travelling Salesman Problem Using Ant Colony Optimization," *International Journal of Computing & Business Research,ISSN* 2229-6166,2012
[4] Zar Chi Su Su Hlaing 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
[5] Marco Dorigo and Luca Maria Gambardella "Ant Colonies for the traveling salesman problem," TR, IRIDIA, 1996-3
[6] T. St¨utzleand H.H. Hoos, "MAX–MIN Ant System," Future Generation Computer Systems, vol. 16, no. 8, pp. 889–914, 2000