ISSN 2250-3153

# Hybrid Ant Colony Optimization and Cuckoo Search Algorithm for Travelling Salesman Problem

**Sandeep Kumar*, Jitendra Kurmi**, Sudhanshu P. Tiwari*****

*M.Tech. Scholar, Lovely Professional University, Jalandhar(Punjab), India
**M.Tech. Scholar, Lovely Professional University, Jalandhar(Punjab), India
***Assistant Professor, Lovely Professional University, Jalandhar(Punjab), India

***Abstract-*** Travelling salesman problem (TSP) is one of the most popular real world combinatorial optimization problem in which we have to find a shortest possible tour that visits each city exactly once and come back to starting city. It ranges among NP hard problem so it is often used as a benchmark for optimization techniques. In this paper a hybrid of Ant Colony Optimization (ACO) and Cuckoo Search (CS) algorithm is proposed for travelling salesman problem. ACO is good metaheuristic algorithm but drawback of this algorithm is that, the ant will walk through the path where the chemical substances called pheromone density is high. It makes the whole process slow hence CS is employed to carry out the problem of local search of ACO. Cuckoo search uses single parameter apart from the population size because of this reason it works efficiently and performs local search more efficiently. The performance of new hybrid algorithm is compared with ACO. The result shows that new hybrid algorithm is better and efficient than simple ACO algorithm**.**

***Index Terms-*** Travelling Salesman Problem, Ant Colony Optimization, Cuckoo Search, Mantegna Search, Egg Laying Region, Probability Parameter, Pheromone Evaporation Coefficient, Pheromone Deposit Factor

## I. INTRODUCTION

Travelling Salesman Problem (TSP) [1] is widely studied problem in Computer Science in which task is to find a hamiltonian path with minimum cost. The TSP is known to be NP-hard. It means that no known algorithm is guaranteed to solve all TSP instances optimality within reasonable execution time. So in addition to find exact solution approaches, various heuristics and metaheuristics algorithms have been developed to solve problems approximately. They are used to find good quality solutions within reasonable execution times. Metaheuristics are normally improvement algorithms, *i.e.*, they start with one or more feasible solutions to the problem at hand and suggest methods for improving such solutions.

ACO [2],[3],[4],[5] is the population based optimization technique which was first introduced by Marco Dorigo, as Ant System in 1996. It was refined as Ant Colony Optimization in 1999 and used to solve the travelling salesman problem. It is inspired by behavior of ants in finding shortest paths from the colony to its food source. Cuckoo search [6],[7],[9] is one of the evolutionary technique which was introduced in 2009 by Yang and Deb . This algorithm is inspired by the egg laying behavior of the cuckoo bird. Cuckoo search is simple and takes less no. of parameters as compared to other metaheuristic algorithms. The major disadvantage of ACO is that the local search it performs is not much faster, so cuckoo search is employed to perform the local search of ACO. In this paper a hybrid of ACO and CS are formulated to solve the travelling salesman problem.

## II. CUCKOO SEARCH

Cuckoo search [6],[7],[9] is a new metaheuristic optimization algorithm. Cuckoo bird does not make nests for laying its egg, it uses other bird's nest for laying eggs. This optimization algorithm is inspired by this behaviour of cuckoo bird. If host bird recognizes the eggs of cuckoo bird, host bird will either destroy the eggs or leave the nest. Each cuckoo can lays its egg in the nest of host bird within a specific region; this region is called Egg Laying Region (ELR) [7]. The working concept of Cuckoo search is this behaviour of cuckoo bird and it is used to solve different optimization problems. The main advantage of cuckoo search is that it uses very less number of parameters which makes it better than other metaheuristic algorithms. Each egg in a nest shows a solution, and a cuckoo egg shows a new solution. Our goal is to find new solution (cuckoos) among all the existing solutions present in the nest.

CS is works on following three rules:

1. Each cuckoo can lay one egg at a time, and can dump its egg in any random selected nest.
2. The nest which would have high quality of eggs will called best nest ant it will go to the next generation;
3. The number of available host nests is fixed and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in (0,1)$.

In Cuckoo Search, leaving the population size n, there is only one parameter $p_a$ and the convergence rate is not related to the parameter $p_a$. It means we do not need to change these parameters for a specific problem. The major advantage of CS is that it is more robust and generic. For number of optimization problems, when it is compared with other metaheuristic algorithms; it performs better.

In cuckoo search main component is Mantegna algorithm which is used to find the best random points in the space. It is used to generate random numbers on the basis of symmetric levy stable distribution. The value of the distribution's parameter $\alpha$ lies from 0.3 to 1.99. The Mantegna algorithm has following phases:

First is to calculate

$$v = \frac{x}{|y|^{1/\alpha}} \quad (1)$$

Here x and $y$ are normally distributed variables with standard deviations, respectively. The value of variable x and y is calculated with the help of following formula:

$$\sigma_x = \left[\frac{\Gamma(1+\alpha) * \sin(\pi\alpha/2)}{\Gamma(\frac{1+\alpha}{2}) * \alpha * 2^{(\alpha-1)/2}}\right]^{1/\alpha} \quad (2)$$

$$\sigma_{y=1} \quad (3)$$

In this way with the help of Mantegna algorithm best points in the space is found.

## III. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) [11],[12] algorithm is a metaheuristic optimization technique that is used to minimize the cost and maximize the efficiency of an optimization problem. ACO is a natural computational technique for which main basis is behaviour of ants. Ant colony optimization is inspired by the foraging or food searching behaviour of real ant colonies. Ants are naturally blind, deaf & dumb so when they move, leave pheromone (chemical material) [8],[10],[13],[14] in its path. On the basis of this pheromone density other ants follow the path. This simulated pheromones attract ants to best trail in the graph because ants choose the path based on the pheromone density. This natural behaviour of artificial ant colonies does not search the exact solutions but provides approximate solutions to continuous and discrete optimization problems, like travelling salesman problem, vehicle routing problem, job scheduling problem etc. ACO uses four main parameters [11] that are influence of pheromone on direction (α)[15],[16], influence of adjacent node distance (β), pheromone evaporation coefficient (ρ) and pheromone depositing factor (Q)[17],[18],[19]. This algorithm updates the trail on the basis of current pheromone density.

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 on city i and constructs the partial solution, the probability moving to the next city j neighbouring on city i is given by

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta} & \text{if } k \in allowed_k \\ 0 & otherwise \end{cases}$$

(4)

where $\tau_{ij}$ is the intensity of trails between edge (i.j) and $\eta_{ij}$ is the heuristic visibility of edge (i, j), and $\eta_{ij}=1/d_{ij}$. α is pheromone influence factor, β is local node influence, and $J_k(i)$ is a set of cities which remain to be visited when the ant is at city $i$. After each ant completes its tour, the pheromone amount on each path will be adjusted with following equation.

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

where ρ is pheromone evaporation coefficient and

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

$$\Delta\tau_{ij}^k(t) = \begin{cases} \dfrac{Q}{L_k}, & \text{if (i, j)} \in \text{tour done by ant k} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

(1- ρ) is the pheromone decay parameter (0<ρ<1) [19] 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.

## IV. TRAVELLING SALESMAN PROBLEM

Given a set of cities and the cost between each pair of them, the travelling salesman problem, is to find the cheapest way of visiting all of the cities and return to starting point. In the standard version we study, the travel costs are symmetric in the sense that travelling from city X to city Y costs just as much as traveling from Y to X.

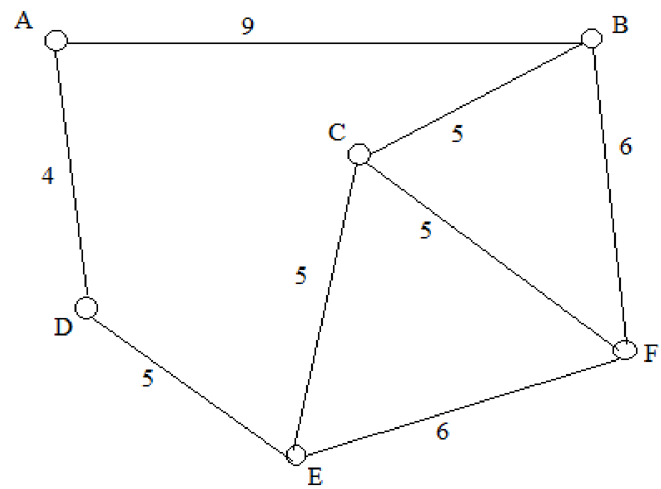


**Fig.1: Graphical Representation of TSP**

In figure 1 each node in graph represents a city and each weighted edges in graph represents path from one city to other city. Weight is the cost to travel to reach city. Fitness function of the TSP will be

$$\mathbf{T_c} = \sum_{k=1}^{d} \mathbf{C_{ij}} \quad (7)$$

where $C_{ij}$ is cost associated of path $i$ to $j$ and $k$ is for the number of city or dimension.

Assumptions for the Travelling Salesman Problem are:
1. There should be finite no. of cities.
2. Each city should connect to next city.

Constraints for the Travelling Salesman Problem are:
1. Each city should traverse exactly once.

2. Salesman should reach at starting city after traversing all the cities.

## V. PROPROSED METHODOLOGY

In this paper we have proposed a solution for Travelling Salesman Problem using ant colony optimization algorithm with cuckoo search. For solving any optimization problem we have to first formulate the problem according to optimization problem. In this case first we formulate the Traveling salesman problem according to our proposed ACO with CS. Our proposed algorithm combines the advantages of Ant Colony Optimization and Cuckoo search. The major drawback in the ACO is that while trying to solve the combinatorial optimization problems the search has to perform much faster, but in ACO ant walk through the path where the chemical substance called pheromone density is high. In order to overcome this drawback, Cuckoo search is used.

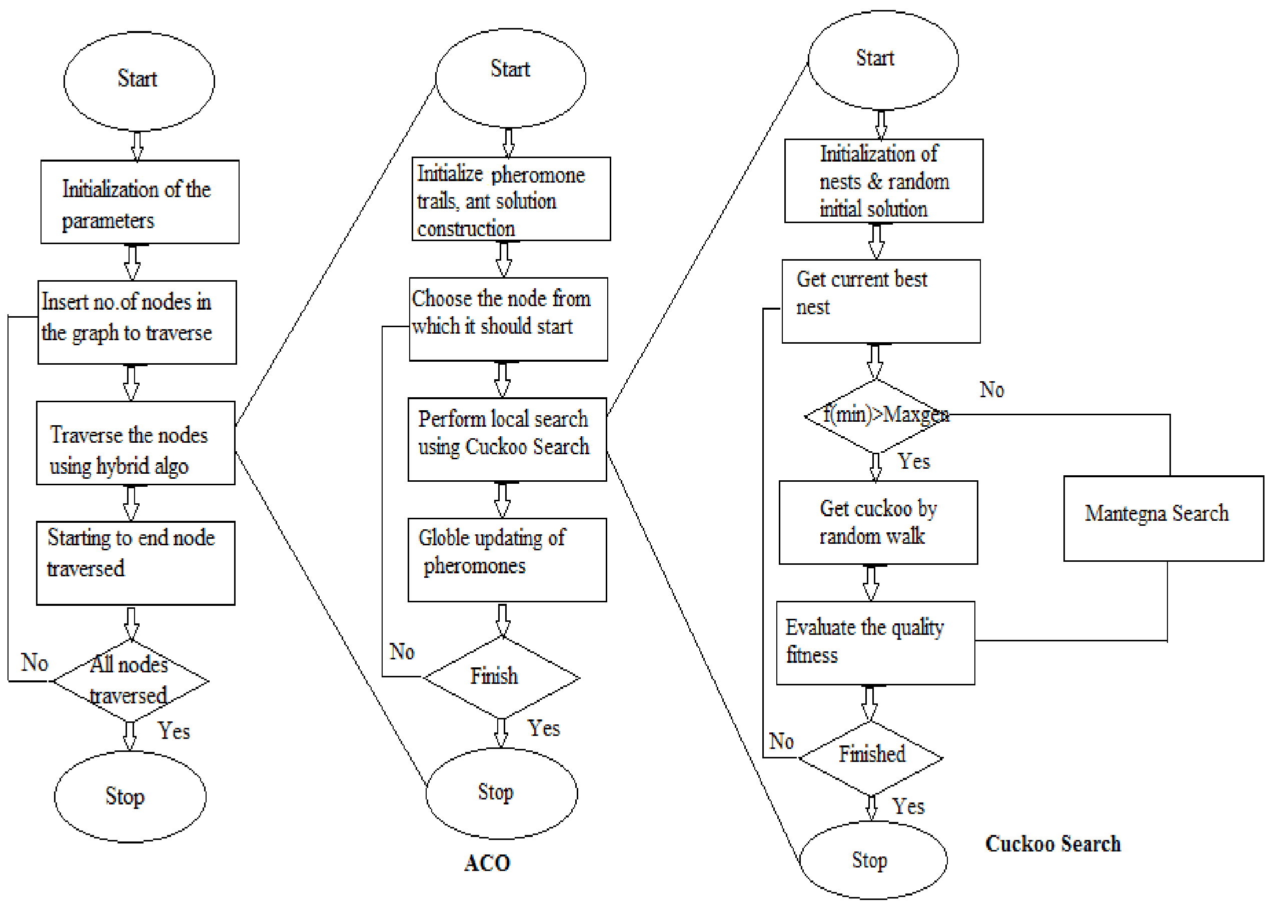


**Fig. 2: Flow Chart of Hybrid Algorithm to solve TSP**

Figure shows the flow chart of the Hybrid algorithm. Firstly initialization of the parameters need to be performed. Then the number of nodes that has to be traverse is initializing. Traversing of nodes are carried out with the help of Hybrid algorithm which uses ACO and Cuckoo search.

*ACO flow chart description:* Initialization of the pheromone trails, and ant solution construction is performed. Each ant is assign to an individual node, assigning of the resource to the task is performed by Cuckoo search. Global updating is performed using the pheromone global update. Termination of task is carried out till the traversal of all the nodes is completed.

*Cuckoo search flow chart description:* Initialization of the nests and the random initial solution is performed. The current best nest is chosen by the random walk and then the evaluation of the quality is fitness is performed. Else apply Mantegna search for the evaluation. Execution is carried out till all the solution is constructed.

The steps of complete proposed hybrid algorithm for TSP is given as:

**Hybrid ACO and Cuckoo Search Algorithm to solve TSP**

1. Initialization.
2. Insert no. of nodes in the graph to traverse.
3. Traverse each node using hybrid algorithm (ACO and Cuckoo Search).
4. Assign each ant to different node /*ACO algorithm starts*/
5. Perform local search using cuckoo search
6. Initialize nest /*Cuckoo Search algorithm starts*/
7. Get the current best nest
8. **While** ($F_{min}$ > max generation)

9. Get the cuckoo by random walk, if not replace it by Mantegna search
10. **End while**
11. Evaluate the quality fitness, randomly choose nests among n (call as j)
12. If ($F_i < F_j$) then replace j value by new solution.

Else Retain the best solution and nest.

13. Choose next city according to fitness value of cuckoo search /*Cuckoo Search algorithm ends*/
14. For each ant update the value of pheromone and calculate fitness function.
15. If total iteration < maximum iteration then go to step 3

Else terminate /*ACO algorithm ends*/

16. Traverse each node exactly once and return back to starting node.
17. If all nodes traversed then terminate

Else repeat step 3 to 16.

18. Stop

## VI. EXPERIMENTAL SETUP AND RESULTS

### A. *Experimental Setup*

For every algorithm there are some control parameters which are used for its efficient working. Hence, there are some control parameters for ant colony optimization algorithm with cuckoo search. We did an extensive literature survey to determining the values of these control parameters. From this we found that the values which we have taken in this experiment are standard values and they are also suitable for this experiment. The control parameters of ACO are $\alpha$ (influence of pheromone on direction), $\beta$ (influence of adjacent node distance), $\rho$ (pheromone evaporation coefficient) and Q (pheromone depositing factor) and values is taken as 1, 2, 0.5 and 2.0 respectively. In cuckoo search value of $p_a$ is taken from 0 to 1. Foe mantegna search the value of alpha is taken from 0.3 to 1.99. The next parameter in our experiment is maximum number of population and we have taken its value to be 5 to 30. No. of ants is taken according to no. of cities, it is same as no. of cities. All results is taken on Windows platform with framework Matlab R2013a.

### B. *Experimental Result*

In this section we analyse the result obtained by our proposed hybrid algorithm. To test the efficiency of our algorithm; results of our hybrid ACO and CS is compared with real simple ACO algorithm results. In TSP we already have the information about the number of cities. We have analysed time taken by proposed hybrid algorithm and simple ACO algorithm to traverse each city exactly once and back to source city. We conducted the experiment by changing the number of cities and then compared our results with simple ACO algorithm. The time is measured in seconds and for graph we have taken symmetric distance in which the distance between two cities is taken from 0 to 19. Results are shown in table 1.

**TABLE I TIME TAKEN BY ACO AND PROPOSED HYBRID ALGORITHM TO SOLVE TSP FOR DIFFERENT NUMBER OF CITIES**

| **No. of cities** | **Time taken by ACO** | **Time taken by hybrid ACO and CS** |
|---|---|---|
| 5 | 1.1572 | 0.9499 |
| 10 | 5.7805 | 3.9142 |
| 15 | 14.9135 | 9.5106 |
| 20 | 30.4802 | 17.9556 |
| 25 | 53.6349 | 27.7136 |
| 30 | 88.2881 | 43.1170 |

The above comparison of results of proposed hybrid algorithm and ACO algorithm is shown in following bar chart:

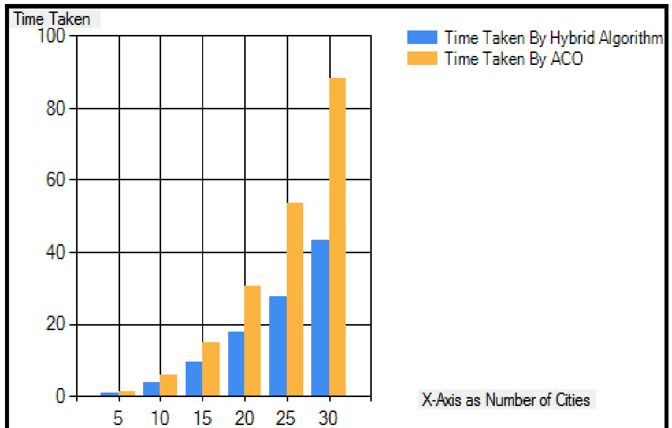


Fig. 3: Comparison of Time Taken By Proposed Hybrid Algorithm and ACO Algorithm

## VII. CONCLUSION

It can be concluded from the results TABLE I that the performance of proposed hybrid algorithm is better than ACO algorithm in most cases because proposed hybrid algorithm performs local search faster than simple ACO algorithm. The main advantage of Cuckoo search is that it takes only one parameter $p_a$ (probability function) with population size. Very less number of parameters makes Cuckoo Search algorithm so efficient and simple than other metaheuristic algorithm. By merging ACO with Cuckoo Search it does not depend on pheromone density to choose the next city which will make the process faster. As future work we have the intention to apply other types of nature inspired algorithms to the Travelling salesman problem.

## ACKNOWLEDGMENT

I would like to express my heartfelt thanks to all my teachers for their valuable advice, helpful comments, kindness support, and precious time for my research.

## REFERENCES

[1] Rajesh Matai1. Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches.Traveling Salesman Problem, Theory and Applications, 1-24 (2011).

[2] Ivan Brezina Jr. Zuzana Cickova, "Solving the Travelling Salesman Problem Using the Ant Colony Optimization, Management Information Systems",Vol No. 4, pp. 010-014, 2011.

[3] Zar Chi Su Su Hlaing and May Aye Khine, "An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem" in: International Conference on Information Communication and Management IPCSIT vol.16, 2011.

[4] Xin-She Yang and Suash Deb, "Cuckoo Search via Levy Flights", in: IEEE, 978-1-4244-5612-3, 2009.

[5] Ramin Rajabioun, "Cuckoo Optimization Algorithm", in: Applied Soft Computing , pp.5508–5518, Elsevier 2011.

[6] S. Nonsiri, S. Supratid, "ModifYing Ant Colony Optimization," IEEE Conference on Soft Computing in Industrial Applications, 2008,pp. 95-100.

[7] Aziz Ouaarab,Belaid Ahiod and Xin-She Yang, "Discrete cuckoo search algorithm for the travelling salesman Problem", in: Springer-Verlag London, vol 24, pp. 1659–1669, 2014.

[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 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

[10] Marco Dorigo. "Ant Colonies for the Traveling Salesman Problem". IRIDIA, UniversitéLibre de Bruxelles. IEEE Transactions on Evolutionary Computation, 1(1):53–66. 1997.

[11] Hui Yu, "Optimized Ant Colony Algorithm by Local Pheromone Update", in: TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.12, No.2, pp. 984 - 990, February 2014.

[12] Isa Maleki and Seyyed Reza Khaze, "A New Solution for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Chaos Theory", in: International Journal of Advanced Research in Computer Science, Volume 3, No. 7, ISSN No. 0976-5697, Nov-Dec2012.

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

[14] Z. A. Othman, H. Md. Rais, and A. R. Hamdan," Strategies DACS3 increasing its performances," European Journal of Scientific Research, 2009.

[15] Z. A. Othman, H. Md. Rais, and A. R. Hamdan, "DACS3: Embedding individual ant behavior in ant colony system," World Academy of Science Engineering and Technology 44 2008.

[16] H. Md. Rais, Z. A. Othman, and A. R. Hamdan, "Improvement DACS3 searching performance using local search," Conference on Data Mining and Optimization, IEEE, 27-28 October 2009.

[17] L. Li, S. Ju, and Y. Zhang, "Improved ant colony optimization for the traveling salesman problem", international conference on intelligent computation technology and automation, IEEE, 2008.

[18] X. Wei, "Improved ant colony algorithm based on information entropy," International Conference on Computational and Information Sciences, IEEE, 2010, 987-0-7695-4270-6.

[19] K. S. Hung, S. F. Su, and S. J. Lee, "Improving ant colony optimization for solving traveling salesman problem," Journal of Advanced Computational Intelligence and Intelligent Informatics,2007.

## AUTHORS

**First Author** – Sandeep Kumar, M.Tech, Lovely Professional University, sandeep.cs12@gmail.com.
**Second Author** – Jitendra Kurmi, M.Tech, Lovely Professional University, jitendrakurmi458@gmail.com.
**Third Author** – Sudhanshu P. Tiwari, Assistant professor, Lovely Professional University, sudhanshu.iiitg@gmail.com