International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 12 3966 - 3971

An Ant Colony Optimization approach to solve Travelling Salesman Problem

Dr. K. Shyamala[1],
Associate Professor,
Dr. Ambedkar Government Arts College (Autonomous),
Chennai.
Shyamalakannan2000@gmail.com

S.Sudha Prabha[2]
Research Scholar,
Dr. Ambedkar Government Arts College (Autonomous),
Chennai.
Sdh.prabha@gmail.com

***Abstract*—**Ant Colony Optimization algorithms (ACO) are meta-heuristic algorithms inspired from the cooperative behavior of real ants that could be used to achieve complex computations and have been proven to be very efficient to many different discrete optimization problems. One such problem is the Travelling Salesman Problem (TSP) which belongs to the class of NP-hard problems, which means that there is no exact algorithm to solve it in polynomial time. The importance of this problem appears in many application areas such as telecommunications, electronics, logistics, transportation, astronomy, industry and scheduling problem. Many algorithms had been proposed to solve TSP each with its own merits and demerits. In this paper, we propose a theoretical overview of TSP and ACO and also an implementation of how ACO can be used to solve TSP.

Keywords--*ACO, NP-hard, TSP, Optimization techniques*

*****

I. INTRODUCTION

Social insects such as ants, termites, wasps, and bees live in almost every land habitat on earth. An ant colony optimization (ACO) technique is an optimization technique to solve combinatorial optimization problems [1]. Swarm intelligence is a relatively new approach to Problem solving that takes inspiration from the social behaviors of insects and of other animals. The attempt in the research of computer technology is to develop algorithms inspired by insect behavior to solve optimization problems. Ant colony optimization (ACO) is one of the most successful techniques in the wider field of swarm intelligence. ACO are multi agent system in which the behavior of each singe agent called ant, is inspired by the behavior of real ants. Ants use the environment as a medium of communication. They exchange information indirectly by deposits pheromones, all detailing the status of their “work”. ACO algorithm models the behavior of real ant’s colonies in establishing the shortest path between food sources and nests. Ants can communicate with one another through chemicals called pheromone 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 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 [2] [3] [4]. Recently, the ACO meta-heuristic has been proposed to provide a common framework to describe and analyze all these algorithm inspired by the same shortest path behavior of ant colonies and the ACO meta-heuristic was defined as a posterior as the result of synthesis effort effectuated on the study of the characteristics of all these ant inspired algorithms and on the abstraction of their common traits [5] [6].

All these studies have contributed to the improvement of the ACO to some extent, but they have little obvious effect on increasing the convergence speed and obtaining the global optimal solution. The ACO algorithm has been applied to the Travelling salesman problem (TSP) which is the problem of finding a shortest closed tour which visits all the cities.

In this paper we give an overview of ACO algorithm for TSP. we first introduce the Ant Colony Optimization Approach in section II and the Ant system in section III. The Travelling Salesman Problem is discussed in section IV. Results are analyzed in section V. The Conclusion is given in section VI.

II. THE ANT COLONY OPTIMIZATION APPROACH

Ant Colony Optimization (ACO) algorithms are constructive stochastic search procedures that makes use of pheromone model and heuristic information on the problem being tackled in order to probabilistically construct solutions. A pheromone model is a set of so-called pheromone trail parameters. The numerical values of these pheromone trail parameter reflect the search experience of the algorithm. They are used to bias the solution construction over time towards the region of the search space containing high quality solutions. The stochastic procedure in ACO algorithms allows the ants to explore a

ISSN: 2321-8169
3966 - 3971

much larger number of solutions; meanwhile, the use of heuristic information guides the ants towards the most promising solutions. The ants search experience is to influence the solution construction in future iterations of the algorithm by a reinforcement type of learning mechanism.

The Real ants can indirectly communicate by pheromone information without using visual cues and are capable of finding the shortest path between food sources and their nests. The ant deposits pheromone on the trail while walking and the other ants follow the pheromone trails with some probabilities which are proportioned to the density of the pheromone. The more ants walk on a trail, the more pheromone is deposited on it and more and more ants follow the trail. Through this mechanism, ants will eventually find the shortest path. Artificial ants imitate the behavior of real ants how they foraging the food, but can solve much more complicated problems than real ants can. A search algorithm with such concept is called Ant Colony Optimization.

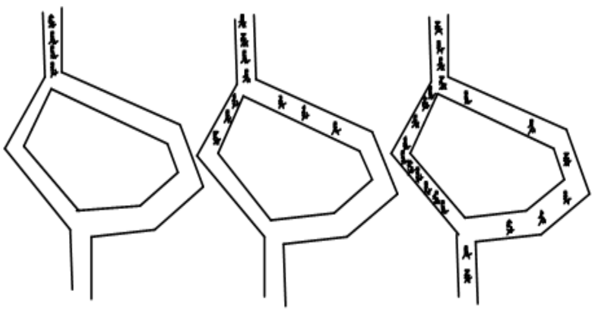

Figure 1. Ants use pheromone as indirect communication to build best tour and show how the ants find the shortest path.

When choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. 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 quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. It has been shown in Figure 1, that the indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources.

III ANT SYSTEM

Ant System (AS) was proposed as the first ACO algorithm for the well known traveling salesman problem (TSP). Despite AS was not competitive with state-of-the-art algorithms on the TSP, it stimulated further research on algorithmic variants for better computational performance. Several improved ACO algorithms for NP-hard problems have been proposed in the literature. Ant Colony System (ACS) and MAX–MIN Ant System (MMAS) algorithm are among the most successful ACO variants in practice. For providing a unifying view to identify the most important aspects of these algorithms, Dorigo et al. [7] put them in a common framework by defining the Ant Colony Optimization (ACO) meta-heuristic. The outline of the ACO Meta heuristic is shown in Algorithm 1.

Algorithm 1 Outline of ant colony optimization Meta heuristic

```
Set parameters, initialize pheromone trails
While termination criterion not satisfied
 Do
         ConstructAntSolution
        ApplyLocalSearch /*optional*/
       Update pheromones
End while
```

It is important to note that the trail update only applies to the arcs of the global-best tour, not to all the arcs like in AS. The parameter ρ again represents the pheromone evaporation. In ACS, only the global best solution receives feedback. Initially, also using the iteration best solution was considered for the pheromone update. Although for smaller TSP instances, the difference in solution quality between using the global-best solution or the iteration-best solution is minimal, for larger instances, the use of the global-best tour gives far better results.

Tour construction: In ACS, ants choose the next city using the pseudo random proportional action choice rule: When located at city i, ant k moves, with probability q0, to city l for which $tu(t).[\eta u]\beta$ is maximal, that is, with probability q0 the best possible move as indicated by the learned pheromone trails and the heuristic information is made (exploitation of learned knowledge). With probability (1 - q0) an ant performs a biased exploration of the arcs according to following equations.

Global pheromone trail update: In ACS, only the global best ant is allowed to add pheromone after each iteration, thus, the update according to equation is modified to.

$$\varsigma(r, s) = (1-\alpha).\varsigma (r, s) + \alpha .\varsigma$$

As algorithm indicates that after the parameter and pheromone (a chemical substance that real ants deposit for local and indirect communication termed as stigmergy), trail initialization on the problem search space, each ant constructs its solution with the probabilistic selection policy and updates pheromone level on the graph according to their solutions. Additionally, local search methods can be applied on the each solution of ants for performance improvement [9]. Detailed description of these three steps of algorithm is:

- a State Transition Rule which brings the concrete ant from a node to another across an arc;
- a Local Updating Rule which updates the pheromones deposited by the ant on the arc it walked in;
- a *Global Updating Rule* which updates the pheromones deposited on the arcs when an ant ends its tour;

$$S = \begin{cases} \arg\max \left\{ [\tau(r,u)].[\eta(r,u)]^{\beta} \right\} & \text{if } q \le q0 \\ S \text{ (selected by using b) otherwise} \end{cases} \quad (1)$$

$$P_k(r,s) = \begin{cases} \dfrac{[\tau(r,u)].[\eta(r,u)]^{\beta}}{\sum_{u \in J_k} [\tau(r,u)].[\eta(r,u)]^{\beta}} & \text{if } s \in J_k \end{cases} \quad (2)$$

$$\tau(r,s) = (1-\rho)\cdot\tau(r,s) + \rho\cdot\Delta\tau(r,s) \quad (3)$$

$$\tau(r,s) = (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau$$

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1} & \text{if } (r,s) \in \text{global– best-tour Where} \\ 0 & \text{Otherwise} \end{cases} \quad (4)$$

The above table shows set of rules of ACS as described in (Dorigo, M.Gambardella), (1) and (2) compose the State Transition Rule; (3) is the Local Updating Rule; (4) is the Global Updating Rule; J_k is the set of nodes not yet visited by the ant k.

AntColony4TSP implements in the method *globalUpdatingRule* the rule **(4)** in table and defines in the member variable *A* the value of the parameter α is the implementation of such a method in java.

AntTSP implements in the method *stateTransitionRule* the *State Transition Rule* mixing the rule **(1)** and **(2)** in Picture for while in the method *localUpdatingRule* it implements the rule **(3)**. In the member variables *B*, *R*, Q_0 *it* defines respectively the values for the parameters β, ρ, *q0*. Is the implementation of both methods in java [8] [9].

After initializing parameters and pheromone trails, the Meta heuristic iterates over three phases: At each iteration, a number of solutions are constructed by the ants; these solutions are then improved through a local search (this step is Optional), and finally the pheromone trails are updated. In ACO for combinatorial problems, the pheromone values are associated with a finite set of discrete values related to the decisions that the ants make. This is not possible in the continuous and mixed continuous-discrete variables cases. Thus, applying the ACO Meta heuristic to continuous domains is not straightforward.

The simplest approach would be to divide the domain of each variable into a set of intervals. A real value is then rounded to the closest bound of its corresponding interval in a solution construction process. This approach has been successfully followed when applying ACO to the protein legends docking problem. However, when the domain of the variables is large and the required accuracy is high, this approach is not viable. Except this approach, there have been some other attempts to apply ACO-inspired algorithms to continuous optimization problems. The proposed methods often took inspiration from some type of ant behaviors, but did not follow the ACO Meta heuristic closely. For this reason, An ACO-inspired algorithm named ACO was proposed, which can handle continuous variables natively. ACO is an algorithm that conceptually directly follows the ideas underlying the ACO Meta heuristic.

Proposed Algorithm

The proposed algorithm combined well distribution strategy of initial ants 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 node must not be visited by the ant) according to A local updating rule (3) is applied

Until ant k has completed a tour

End for

Local search is applied to improve tour

A global updating rule is applied

Compute entropy value of current pheromone trails

Update the heuristic parameter

Until end_condition

End

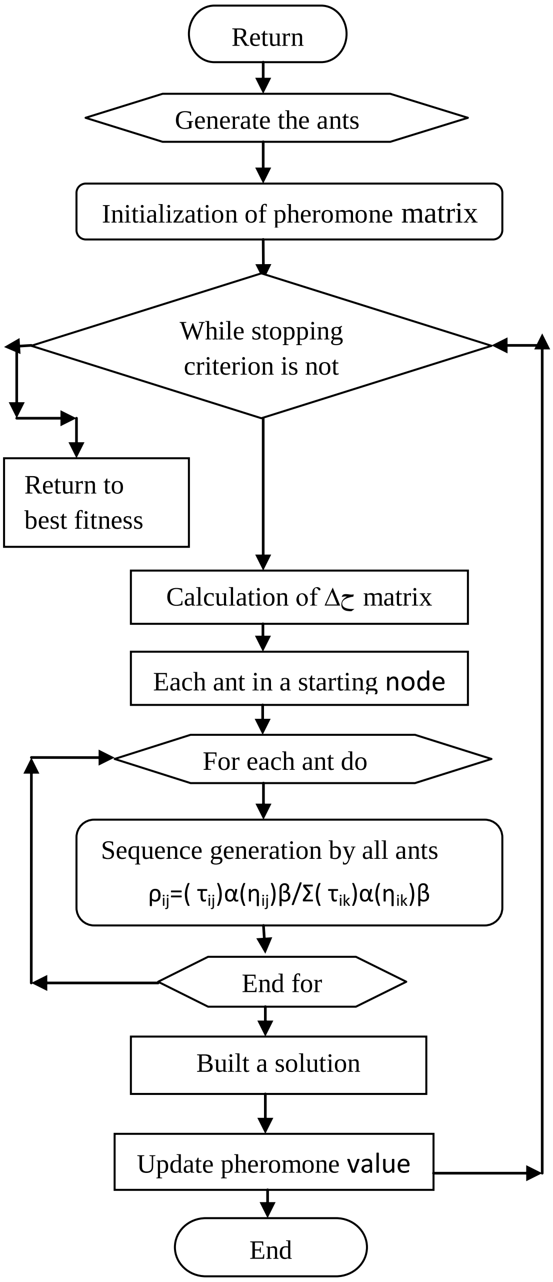

Figure 2. Ant Colony Optimization Algorithm

IV. TRAVELLING SALESMAN PROBLEM

Travelling salesman problem (TSP) is a well known, popular and extensively studied problem in the field of combinatorial optimization and attracts computer scientists, mathematicians and others. Its statement is deceptively simple, but yet it remains one of the most challenging problems in operational research. It is also an optimization problem of finding a shortest closed tour that visits all the given cities. It is known as a classical NP-complete problem, which has extremely large search space and is very 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. TSP is an NP-hard problem and researchers especially mathematicians and scientists have been studying to develop efficient solving methods since 1950's. Graph theory defines the problem as finding the Hamiltonian cycle with the least weight for a given complete weighted graph. The traveling salesman problem is widespread in engineering applications. It has been employed in designing hardware devices and radio electronic devices, in communications, in the architecture of computational networks, etc. In addition, some industrial problems such as machine scheduling, cellular manufacturing and frequency assignment problems can be formulated as a TSP [10].

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:

$$dij = \sqrt{(xi - xj)^2 + (yi - yj)^2}$$

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 40 cities. 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.

V. ANALYSIS AND RESULTS

The computational results in this technique with 30 nodes show that MMAS, in general, is able to find very high quality solutions for all instances; furthermore, for almost all instances, MMAs finds the optimal solution in at least one of the runs. This is an encouraging result which shows the viability of the ant approach to generate very high quality solutions for finding optimal path. According to these results, MMAS is currently the best performing ant approach for the route finding [11].

To test accuracy of algorithm implementations, we compared them on some limits of TSPLIB instances. Tests are executed over the trails with the parameters for MMAS. The computational result shows that generally MMAS achieve best performance. Regarding the performance of AS, it can be seen that AS performs so poorly compared to other algorithms. Additionally, we can say that additional reinforcement optimization techniques (MAX-MIN ant system) implemented over the TSP really increase the AS performance.

Figure 3. Iteration runs of ant System

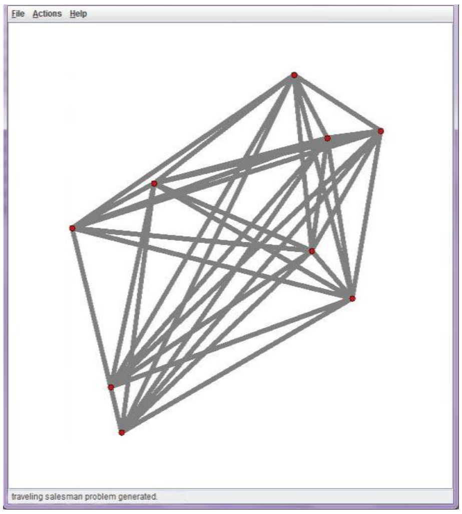

Figure 4. Ant cycle for TSP

The implementation of the proposed algorithm is done using java language. Iteration runs of ant system are shown in figure 3. Figure 4 shows the Ant cycle for travelling salesman problem and also many ways TSP can be improved to reach the needs of the tour. Path movement of ant system is shown in figure 5. First random Traveling salesman nodes are generated. Next Ant colony will be created. Finally the shortest path will be selected.

VI. CONCLUSION

This Paper presents the ant colony optimization approach for solving travelling salesman problem and is an attempt to study and analyze its performance.

The experimental results shows that ACO algorithm provides relatively good results by a comparatively low number of iterations, and is therefore able to find an acceptable solution in a comparatively short time. But there are many ways ACO can be improved so that the number of tours needed to reach a comparable performance level can diminish, making its application to larger problem instances feasible. Several researches are going on and improvements are being made on the ACO algorithms to improve its efficiency in terms of speed and the ability to find better solutions.

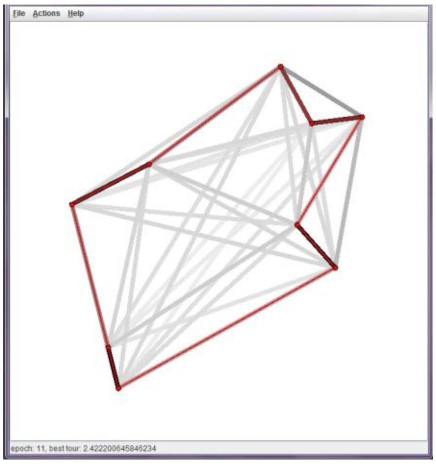

Figure 5: Path movement of ant and shortest path

REFERENCES

[1] Zar Chi Su Su Hlaing, May Aye Khine," *An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem*",2011, InternationalConference on Information Communication and Management IPCSIT vol.16 (2011) © (2011) IACSIT Press, Singapore.

[2] Thomas stautzle and Marco Dorigo,"*ACO Algorithms for the Traveling Salesman Problem*", IRIDIA, Universite Libre de Bruxelles, Belgium, Presently on leave from FG Intellektik, TUDarmstadt, Germany. John Wiley & Sons, 1999.

[3] Hifza Afaq, Sanjay Saini, "*On the Solutions to the Travelling Salesman Problem usingNature Inspired Computing Techniques*", IJCSI International Journal of Computer Science Issues, Vol. 8,Issue 4, No 2, July 2011.

[4] Marco Dorigo, Mauro Birattari, and Thomas Stutzle, "*Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique*", RIDIA – Technical Report Series, Technical Report No.TR/IRIDIA/2006-023,September 2006.

[5] Mohammed Alhanjouri, Belal Alfarra, "*Ant Colony versus Genetic Algorithm based on Travelling Salesman Problem*", Int. J. Comp. Tech. Appl., Vol 2 (3), 570-578.

[6] Megha Aggarwal,Dr. Saroj , " *Compute Travelling Salesman Problem Using Ant ColonyOptimization*", Proceedings of 'I-Society 2012' at GKU, Talwandi Sabo Bathinda (Punjab).

[7] Dorigo and Gambardella, " *Ant colonies for the traveling salesman problem*", Accepted for publication in BioSystems, 1997.

[8] Issmail Ellabib , Paul Calamai , Otman Basir, " *Exchange strategies for multiple Ant Colony System*", Information Sciences 177 (2007) 1248–1264.

[9] Raka jovanovic, Milan tuba, Dana simian, "An *Object-Oriented Framework with Corresponding Graphical User Interface for Developing Ant Colony Optimization Based Algorithms*", ISSN: 1109-2750, Issue 12, Volume 7, and December 2008.

[10] Leonora Bianchi, Luca Maria Gambardella, and Marco Dorigo," *Solving the Homogeneous Probabilistic Traveling Salesman Problem by the ACO Metaheuristic*" IRIDIA, Universite Libre de ,Bruxelles CP 194/6, Avenue Franklin Roosevelt 50, 1050 Brussels, Belgium.

[11] Oloruntoyin Sefiu Taiwo, Ojo Josiah, Amao Taiwo, Salawudeen Dkhrullahi & Oloruntoyin Kehinde Sade," *Implementation of heuristics for solving travelling salesman problem using nearest neighbor and nearest insertion approaches*", International Journal of Advance Research, IJOAR .org, Volume 1, Issue 3, March 2013, Online: ISSN 2320-9194.