# Survey of Finding Solution for optimization problem using Ant Colony Optimization

**B. Sasikala[1], Dr. V. P. Eswaramurthy[2]**

[1]Assistant Professor, Bharathi-dasan University Constitutent College for Women,Orathanadu

[2]Assistant Professor, Government Arts and Science College, Kumarapalayam

**Abstract**: ***Ant Colony Optimization is one of the meta- heuristic algorithms and first member of ACO is Ant System (AS). AS uses a population of co-operating ants also known as agents. The cooperation phenomenon among the ants is called foraging and recruiting behavior. This describes how ants explore the world in search of food sources, then find their way back to the nest and indicate the food source to the other ants of the colony. The nature of ants, that collectively solve hard problems, gave rise to artificial ant algorithms. These algorithms were also proposed as a multi-agent approach in order to solve hard combinatorial optimization problems. ACO meta-heuristic introduces main features of artificial and these features have inspired different ant algorithms to solve hard optimization problems.***

**Keywords:** Ant Colony Optimization (ACO), meta-heuristic, Combinatorial Optimization Problem (COP)

## 1. Introduction

The first ant colony optimization (ACO) called ant system was inspired through studying of the behavior of ants in 1991 by Macro Dorigo and co-workers [1]. An ant colony is highly organized, in which one interacting with others through pheromone in perfect harmony. Optimization problems can be solved through simulating ant's behaviors. Since the first ant system algorithm was proposed, there is a lot of development in ACO. In ant colony system algorithm, local pheromone is used for ants to search optimum result. In principle, ACO can be applied to any discrete optimization problem, for which some solution construction mechanism can be conceived. These applications comprise two main application fields. NP-hard problems, where ACO algorithms are coupled with extra capabilities, such as heuristic information and local search methods. Dynamic optimization problems, in which some properties of the problem change over time concurrently with the optimization process that has to adapt to the problem's dynamics.

## 2. Ant Colony Optimization

Ant colony optimization[2,3] is a meta-heuristic technique that uses artificial ants to find solutions to combinatorial optimization problems. ACO is based on the behavior of real ants and possesses enhanced abilities such as memory of past actions and knowledge about the distance to other locations. In nature, an individual ant is unable to communicate or effectively hunt for food, but as a group, ants possess the ability to solve complex problems and successfully find and collect food for their colony. Ants communicate using a chemical substance called pheromone. The nature of the ants varies considerably. Some types of ants are capable of finding the best route between a food source and the nest. All ants wander back and forth between the food source and the nest after finding the food source. Initially, each ant randomly chooses one of the two bridges. They also deposit pheromone on the ground during the wandering. This pheromone will affect paths of future ants, because the ants have a tendency to follow paths marked with pheromone. If it follows the trail, the ant's own pheromone reinforces the existing trail, and the increase in pheromone increases the probability of the next ant selecting the path. Therefore, the more ants that travel on a path, the more attractive the path becomes for subsequent ants. Additionally, an ant using a short route to a food source will return to the nest sooner and therefore, mark its path twice, before other ants return. After some time, the shorter routes become very strongly marked with pheromone and thus, all ants will be following the shorter routes. The evaporation of pheromone also makes less desirable routes more difficult to detect and further decreases their use. Detailed descriptions of ant behavior relating to ACO are found in [4,5].

## 3. ACO Algorithms

Several ACO algorithms have been proposed in the literature. Here we present the three most successful ones, 1)Ant System(AS) the first implementation of an ACO algorithm followed by 2)MAX-MIN Ant System(MMAS) and 3)Ant Colony System(ACS).

### 1) Ant System

Ant System was the first ACO algorithm to be proposed in the literature [1,6,7]. Its main characteristic is that the pheromone values are updated by all the ants that have completed the tour. The pheromone update for $\tau_{ij}$, that is, for edge joining cities i and j, is performed as follows :

$$\tau_{ij} = (1- \rho ).\ \tau_{ij} + \Delta\tau_{ij} \qquad (1)$$

where $\rho(0<\rho< 1)$, is a user-defined parameter called evaporation coefficient, and $\Delta\tau_{ij}$ represents sum of the contributions of all ants that used move (i, j) to construct their solution and is calculated as given in equation (2).

$$\Delta\tau_{ij} = \sum_{k=1}^{t} \Delta\tau_{ij}^{k} \qquad (2)$$

where $t$ is the total number of ants, $\Delta\tau_{ij}^{k}$ is the amount of trail laid on edge ($i$, $j$) by the k[th] ant and is computed as in equation (3).

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant k uses edge (i, j) in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $Q$ is a constant parameter selected according to size of the problem and $L_k$ is the length of the path traveled by $k^{th}$ ant. When constructing the solutions, the ants in AS traverse a construction graph and make a probabilistic decision at each vertex. The transition probability of the $k^{th}$ ant moving from node $i$ to node $j$ is given by

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{j \in S} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $S$ be a set of nodes at a decision point $i$. $\tau_{ij}$ is the quantity of pheromone on the edge between node $i$ and node $j$. $\eta_{ij}$ is heuristic information on the node $j$. $\alpha$ and $\beta$ are the parameters that control the relative importance of the pheromone versus the heuristic information $\eta_{ij}$ given by

$$\eta_{ij} = 1/d_{ij} \quad (5)$$

where $d_{ij}$ is the length of edge (i, j).

### 2) MAX-MIN Ant System

MAX-MIN Ant System is an improvement over the original AS System. This algorithm was proposed by Stützle and Hoos [8], who introduced a number of changes of which the most important are the following:

- Only the best ant can update the pheromone trails, and
- The minimum and maximum values of the pheromone are limited.

The pheromone update implemented is as follows:

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \Delta\tau_{ij}^{best} \quad (6)$$

where $\Delta\tau_{ij}^{best}$ is the pheromone update value defined by

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/L_{best} & \text{if(i, j) belongs to the best tour} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

$L_{best}$ is the length of the tour of the best ant. This may be(subject to the algorithm designer decision) either the best tour found in the current iteration—*iteration-best*, $L_{ib}$ or the best solution found since the start of the algorithm — *best-so-far*, $L_{bs}$ or a combination of both. The process of pheromone update in MMAS is concluded by verifying that all pheromone values are within the imposed limits:

$$\tau_{ij} = \begin{cases} \tau_{min} & \text{if } \tau_{ij} < \tau_{min} \\ \tau_{ij} & \text{if } \tau_{min} \le \tau_{ij} \le \tau_{max} \\ \tau_{max} & \text{if } \tau_{ij} > \tau_{max} \end{cases} \quad (8)$$

Concerning the lower and upper bounds on the pheromone values, $\tau_{min}$ and $\tau_{max}$, they are typically obtained empirically and tuned on the specific problem at hand[9].Nonetheless, some guidelines have been provided for defining $\tau_{min}$ and $\tau_{max}$ on the basis of analytical considerations[8].

### 3) Ant Colony System

Another improvement over the original AS was Ant Colony System(ACS), introduced by Gambardella and Dorigo [10, 11]. The most interesting contribution of ACS is the introduction of a *local pheromone update* in addition to the pheromone update performed at the end of the construction process (called here offline pheromone update). The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} = (1-\varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \quad (9)$$

where $\varphi \in (0, 1]$ is the pheromone decay coefficient, and $\tau_0$ is the initial value of the pheromone.

The main goal of the local update is to diversify the search performed by subsequent ants during an iteration: by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and, hence, to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration.

The offline pheromone update, similarly to MMAS, is applied at the end of each iteration by only one ant (either the one that found the best solution in the iteration or the best-so-far). However, the update formula is slightly different:

$$\tau_{ij} \leftarrow \begin{cases} (1-\rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij} & \text{if (i, j) belongs to best tour,} \\ \tau_{ij} & \text{otherwise} \end{cases} \quad (10)$$

As in MMAS, $\Delta\tau_{ij} = 1/L_{best}$, where $L_{best}$ can be either $L_{ib}$ or $L_{bs}$.

## 4. Conclusion

In recent years, the interest of the scientific community in ACO has risen sharply. In fact, several successful applications of ACO to a wide range of different discrete optimization problems are now available. The large majority of these applications are to NP-hard problems; that is, to problems for which the best known algorithms that guarantee to identify an optimal solution have exponential time worst case complexity. The use of such algorithms is often infeasible in practice, and ACO algorithms can be useful for quickly finding high-quality solutions. The number of successful applications to academic problems has motivated people to adopt ACO for the solution of industrial problems, proving that this computational intelligence technique is also useful in real-world applications. A better understanding of the theoretical properties of ACO algorithm is certainly another research direction that will be pursued in the future.

## References

[1] Dorigo,M., Maniezzo,V.,and Colorni,A., ―Positive Feedback as a Search Strategy", Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991.

[2] Dorigo, M. and Di Caro, G., ―The ant colony optimization meta-heuristic", in *New Ideas in*

*Optimization*, Corne, D., Dorigo,M.,and Glover,F.,Eds.,McGraw- Hill, New York, 1999, pp. 11–32.

[3] Dorigo, M., Stützle, T., ―Ant Colony Optimization".MIT Press, Cambridge, MA(2004).

[4] Tarasewich P, McMullen PR. ―Swarm intelligence: power in numbers". Commun ACM 2002;45(8):62–7.

[5] Dorigo M, Di Caro G, and Gambardella L.M., ―Ant algorithms for discrete Optimization", Artificial Life, vol. 5, No. 2, pp. 137–172, 1999.

[6] Dorigo, M.,―Optimization, Learning and Natural Algorithms(in italian)", Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy,1992.

[7] Dorigo, M., Maniezzo, V., and Colorni, A., ―Ant System: Optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics—*Part B*, vol. 26, No. 1, pp. 29–41, 1996.

[8] Stützle, T. and Hoos, H.H., ―MAX-MIN Ant System", *Future Generation Computer Systems.* vol 16, No. 8, pp. 889-914, 2000.

[9] Socha, K., Knowles, J., and Sampels, M., ―AMAX-MIN Ant System for the university timetabling problem", Proc. ANTS 2002 3rd Int. Workshop on Ant Algorithms, Lecture Notes in Computer Science, Vol. 2463, Berlin, Germany: Springer Verlag, 2002, p. 1.

[10] Dorigo, M. and Gambardella, L.M., ―Ant Colony System: A cooperative learning approach to the traveling salesman problem", IEEE *Transactions on Evolutionary Computation*, vol. 1, No. 1, pp. 53–66, 1997.

[11] Gambardella, L.M. and Dorigo, M., ―Solving symmetric and asymmetric TSPs by ant colonies", in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC'96)*, T. Baeck *et al.*, Eds. IEEE Press, Piscataway, NJ, 1996, pp. 622–627.