

# SWARM INTELLIGENCE FROM NATURAL TO ARTIFICIAL SYSTEMS: ANT COLONY OPTIMIZATION

O. Deepa<sup>1</sup> and Dr. A. Senthilkumar<sup>2</sup>

<sup>1</sup>Research Scholar, Department of Computer Science,  
Bharathiar University, Coimbatore, Tamil Nadu, India.

<sup>2</sup>Asst. Professor, Department of Computer Science,  
Arignar Anna Government Arts College, Namakkal, Tamil Nadu, India.

## ***Abstract***

*Successful applications coming from biologically inspired algorithm like Ant Colony Optimization (ACO) based on artificial swarm intelligence which is inspired by the collective behavior of social insects. ACO has been inspired from natural ants system, their behavior, team coordination, synchronization for the searching of optimal solution and also maintains information of each ant. At present, ACO has emerged as a leading metaheuristic technique for the solution of combinatorial optimization problems which can be used to find shortest path through construction graph. This paper describe about various behavior of ants, successfully used ACO algorithms, applications and current trends. In recent years, some researchers have also focused on the application of ACO algorithms to design of wireless communication network, bioinformatics problem, dynamic problem and multi-objective problem.*

## ***Keywords***

*Ant colony optimization, biologically inspired algorithm, artificial swarm intelligence, metaheuristic technique, combinatorial optimization*

## **1. INTRODUCTION**

Many researchers tried to solve complex problems to get feasible results. Feasible solutions are those which give all the possible results. Optimization problem is the problem of finding best result from all the feasible results. Many optimization techniques are inspired from natural phenomenon like ant colony optimization, bee colony optimization etc. These techniques are based on *Swarm Intelligence*. The term swarm intelligence is a probabilistic technique for solving computational problems which can be used to get optimal solution [1]. The shortest path behavior of foraging ant colonies and by a similar common process of reverse-engineering of this behavior is at the very root of ACO's design [2]. Therefore, it is the starting point of our description of ACO's genesis and can be summarized as follow.

### **1.1. SOCIAL INSECTS: SWARM INTELLIGENCE**

Insects that live in colonies, like ants, bees, termites and wasps have inspired naturalists as well as poets for many years. "What is it that governs here? What is it that issues orders, foresees the future, elaborates plans and preserves equilibrium?" wrote Maeterlinck. In a social insect colony, a worker usually does not perform all tasks, but rather specializes in a set of tasks according to its

morphology, age or chance. This division of labor among nestmate, whereby different activities are performed simultaneously by groups of specialized individuals. The modeling of social insects by means of self-organization can help design artificial distributed problem-solving devices that self-organize to solve problems – swarm-intelligent systems. The goal of swarm intelligence is the design of multi-agent intelligent systems by taking inspiration from the collective behavior of social insects and other animal societies such as fish schools, flocks of birds and etc.[3].

## 1.2. SPECIAL CHARACTERISTICS OF SOCIAL INSECTS

### 1.2.1. SELF-ORGANIZATION

The discovery that self-organization [4] may be at work in social insects and also provides us with powerful tools to transfer knowledge about social insects to the field of intelligent system design. In effect, a social insect colony is a decentralized problem-solving system, comprised of many relatively simple interacting entities. The daily problems solved by a colony include finding food, building, repairing or extending a nest, efficiently dividing labor among individuals, efficiently feeding the brood, responding to external challenges, spreading alarm, etc. Many of these problems have counterparts in computer science and engineering. One of the most important features of social insects is that they can solve these problems in a very flexible (allows to adaptation to changing environments) and robust way (empower the colony with the ability to function even though some individuals may fail to perform their tasks).

### 1.2.2. THE INDIVIDUAL DISCRIMINATION CAPABILITIES

Many social insect are able to discriminate [5] between nestmates and non-nestmates. In an ant society individuals coming from different groups bear it own chemical identity and those individual present discrimination capabilities between different chemical profiles. However, at the collective level these groups may cooperate and act together. To understand this apparent contradiction, amplification is the main component of many collective phenomena in social insects. Such collective response is a generic property of social phenomena governed by amplification process.

## 2. RELATED WORK

### 2.1. COMBINATORIAL OPTIMIZATION (CO)

Combinatorial optimizations [6] are of high importance both for the scientific world as well as for the industrial world like telecommunication network design, artificial intelligence, shape optimization or computational biology. The research community has simplified many of these problems like traveling salesman problem, minimum spanning tree problem, etc.

Generally CO is a *NP-hard* problem can be defined as  $P = (S, \Omega, f)$  is an optimization problem where  $S$  is a search space defined over a finite set of discrete decision variables,  $\Omega$  is a set of constraints among the variables and an objective function

$$f: S \rightarrow R^+ \quad (1)$$

that assigns a positive cost value to each of the objects  $s \in S$ . The goal is to find an object of minimal cost value. The search space  $S$  is defined as a set of discrete variables  $X_i$  where  $i = 1, 2, \dots, n$  with values  $v_i^j \in D_i = \{v_i^1, \dots, v_i^{|D_i|}\}$ , that is, each assignment of  $X_i$  has a value  $v_i^j$  assigned from its domain  $D_i$ . The set of feasible solutions  $S_\Omega$  is given by the elements of  $S$  that satisfy all the constraints in the set  $\Omega$ . A solution  $s^* \in S_\Omega$  is called a global optimum if and only if

$$f(s^*) \leq f(s) \forall s \in S_\Omega \quad (2)$$

The set of all globally optimal solutions is denoted by  $S_\Omega^* \subseteq S_\Omega$ . Solving a CO problem requires finding at least one  $s^* \in S_\Omega^*$ .

Due to the practical importance of CO problems, many algorithms to tackle them have been developed. These algorithms can be classified as either *complete* or *approximate* algorithms. Complete algorithms are guaranteed to find an optimal solution for every finite size in bounded time. But this algorithm might need exponential computation time in the worst-case. The development of approximate algorithm, in which we sacrifice the guarantee of finding optimal solutions for the sake of getting good solutions in a significantly reduced amount of time.

## 2.2. FROM REAL ANT COLONIES TO THE ANT COLONY OPTIMIZATION

The Ant colony optimization is a metaheuristic technique based on swarm intelligence for solving combinatorial optimization problems developed by Marco Dorigo and his co-researchers at the beginning of the 1990's [7] which is inspired upon stigmergic communication [8] to find out the shortest path. The ant colony algorithm is an algorithm for finding optimal paths that is based on the behavior of ants searching for food which is away from anthill [9]. At first, the ants wander randomly. When an ant finds a source of food, it walks back to the colony leaving a chemical substance on the way called *pheromone* that show the path has food. The pheromones that are laid to and fro the food source is called as *pheromone trail*. It is very useful to find out direction of food source. When other ants come across the pheromone trail, they are likely to follow the path with a certain probability. If they do, then populate the path with their own pheromones. As more ants find the path, it gets stronger pheromone concentrations. Because the ants secreted pheromones every time they bring food. They choose shortest path from many path by using pheromone trail can be conveniently seen in the terms of a *collective learning process* happening over time. Shorter paths are more likely to be stronger, hence optimizing the *solution*. A similar approach can be used find near-optimal solution to the traveling salesman problem. The foraging of an ant colony is simulated as follows: At each iteration all the ants are initially placed in  $v_s$  and then moves to  $v_d$  is illustrated stepwise in Figure 1.

In the ACO algorithm, an artificial ant colony simulates the pheromone trail following behavior of real ants [10]. Artificial ants move on a *construction graph* representing a specific problem to construct solutions successively. The artificial pheromone that corresponds to the record of routes taken by the ant colony is accumulated at run-time through a learning mechanism. Individual ants concurrently collect necessary information, stochastically make their own decisions and independently construct solutions in a stepwise manner. The information required for making a decision at each step includes pheromone concentration, problem related data and heuristic function values. The pheromone laid on the path belonging to the iteration-best solution will be positively increased to become more attractive in the subsequent iterations. Because of self-organize and reverse-engineering behavior, ACO can effectively and efficiently solve a wide class of combinatorial optimization problems.

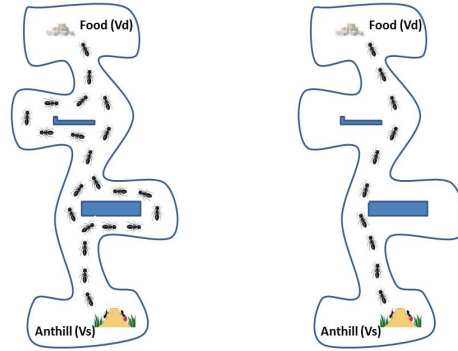


Figure 1. Shortest path finding capability of ant colonies

In dynamic system, the ant colony algorithm works in graphs with different topologies. A graph  $G$  is a pair  $G = (V, E)$ , where  $V$  consists of two nodes, namely  $v_s$  representing the anthill and  $v_d$  representing the food source. Furthermore,  $E$  consists of two links, namely  $e_1$  and  $e_2$ , between  $v_s$  and  $v_d$ . To  $e_1$ , to assign a length of  $l_1$  and  $e_2$  a length of  $l_2$  such that  $l_2 > l_1$ , that is  $e_1$  represents the short path and  $e_2$  represents the long path between  $v_s$  and  $v_d$ .

To introduce an artificial pheromone value  $\tau_i$  for each of the two edges  $e_i$ ,  $i = 1, 2$ . Such a value indicates the strength of the pheromone trail on the corresponding path. Each ant, starting from anthill ( $v_s$ ), an ant chooses with probability between path  $e_1$  and  $e_2$  for reaching the food source ( $v_d$ ). Obviously, if  $\tau_1 > \tau_2$ , the probability of choosing  $e_1$  is higher and vice versa. For returning from  $v_d$  to  $v_s$ , an ant uses the same way and it changes the artificial pheromone value  $\tau_i$  associated to the used path as follows

$$\tau_i = \tau_i + \frac{Q}{l_i} \quad (3)$$

where  $Q$  is a positive constant parameter of the model. In other words, the amount of artificial pheromone that is added depends on the length of the chosen path (shorter path).

Ants deposit pheromone on the paths on which they move. Thus, the chemical pheromone trails are modeled as follows

$$p_i = \frac{\tau_i}{\tau_1 + \tau_2} \text{ where } i = 1, 2 \quad (4)$$

In nature the deposited pheromone is subject to evaporation over time. Moreover, ACO deals with a process in which decreasing in amount of pheromone deposited on every path by the time is known as *trail pheromone evaporation*. To simulate this pheromone evaporation in the artificial model is

$$\tau_i = (1 - p) \cdot \tau_i \quad (5)$$

where  $i = 1, 2$  and the parameter  $p \in (0, 1]$  that regulates the pheromone evaporation. Finally, all ants conduct their return trip and reinforce their chosen path. Above mentioned processes are illustrated in the flowchart for ACO approach is as shown in Figure 2.

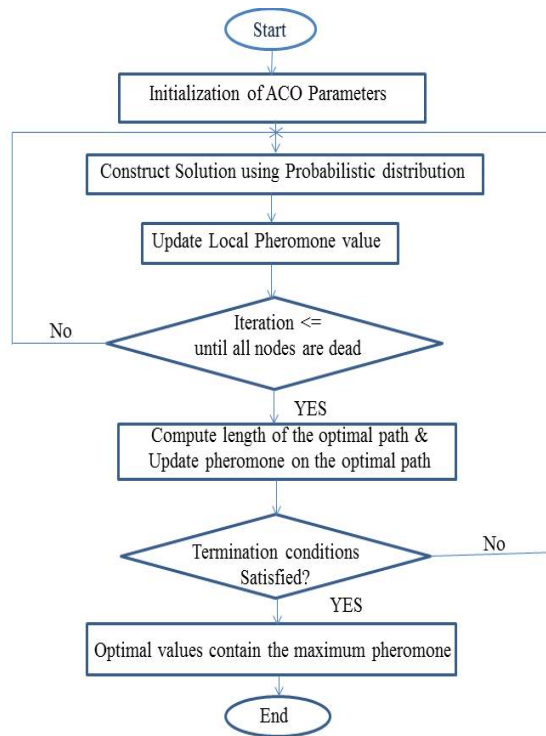


Figure 2. Flowchart of Ant Colony Optimization

### 2.3. ADVANTAGES OF ACO

- ACO adaptive in nature and allow to adaptation to changing environment for dynamic applications.
- It has an advantage of distributed computation and can be possible to prove that it is convergence.
- It gives positive feedback which leads to discovery of good solutions and can be used in dynamic applications and displays powerful robustness.
- In analyses of real dimension networks, to verify the possibility of developing a meta-heuristic algorithm that allow network flows to be calculated more quickly than by using traditional algorithms.
- It allows dynamic rerouting through shortest path if one node is broken. Most other algorithms instead assume that the network is static.

### 2.4. DISADVANTAGES OF ACO

- In standard ACO algorithms, the pheromone trail and the heuristic values depend on the scale of the problem.
- Theoretical analysis is difficult.
- Its convergence is guaranteed but time to convergence is uncertain.
- It is prone to falling in the local optimal solution.

### 3. ACO FOR TRAVELLING SALESMAN PROBLEM

The Travelling Salesman Problem (TSP) also plays an important role in ACO research. The first ACO algorithm, called Ant System, as well as many of the ACO algorithms proposed subsequently, was first tested on the TSP. TSP is one of the most widely known NP-hard problems. In the TSP, we are given a set of cities and the distances for each pair of distinct cities. The salesman has to visit every city exactly one time and then back to start city return to the starting city in the end. Our goal is to find a closed tour with minimal cost. ACO can be applied to the TSP in a straightforward way, as described in the following algorithm is illustrated in Figure 3.

Step 1: Number of artificial ants starts at random cities and wanders randomly through edges

Step 2: Each ant builds a solution based on constraints  
i.e) each city has to be visited exactly once

Step 3: The choice of next random city is based on the pheromones associated with each edge or vertex depending on the problem

Step 4: The probability of choosing next city 'j' from current city 'i' is given by

$$P_{ij}^k(t) = \frac{[\tau_{ij}^\alpha][\eta_{ij}^\beta]}{\sum_{l \in N_i^k} [\tau_{il}^\alpha][\eta_{il}^\beta]}$$

where,  $\tau_{ij}$  represents the pheromone concentration with the edges

$\eta_{ij}$  represents the heuristic information (obtained based on experience)

$\eta_{ij} = 1/d_{ij}$ , where  $d_{ij}$  is the distance between two cities or vertex

here, if  $\alpha = 0$ , probability is based purely on heuristics. Cities closed to each other are chosen

if  $\beta = 0$ , probability is based on pheromone concentration.

Sometimes can lead to localized search space.

Step 5: Pheromone Update: Once all ants travels through each city, pheromones associated with each vertex is updated

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

where ' $\rho$ ' represents the evaporation rate to avoid accumulation of the pheromones

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L_k} \\ 0 \end{cases}$$

where  $L_k$  is the total length of the ant's 'k' tour

Figure 3. ACO for Travelling Salesman Problem

## 4. STANDARD ACO ALGORITHMS

### 4.1. ANT SYSTEM (AS)

The first ant colony algorithm is Ant System to be developed by Dorigo, Maniezzo and Colorni, an algorithm for the traveling salesman problem [12]. The AS was designed as multi-agent system for combinatorial optimization. Its agents were called ants and were using a probabilistic decision rule, while the learned qualities of decision variables were indicated with the term pheromone. In AS, ants construct solutions based on two components: pheromone trails and problem-dependent heuristic information. These algorithms have suffered frequent modifications in order to improve their efficiency. There are several improvements on ant system. The first improvement of AS, called the elitist strategy for ant system (EAS) [13], because each ant that finds a better solution has the chance to deposit more pheromone.

#### 4.2. ANT COLONY SYSTEM (ACS)

The enhanced version of Ant system is Ant colony system to be proposed by Dorigo and Gambardella [14]. The important difference between ACS and AS is the form of the decision rule used by the ants during the construction process. It consists of using two pheromone update rules, local and global. The pheromones are updated locally during solutions construction and a global pheromone update occurs at the end of the construction process. Ants in ACS use the *pseudorandom proportional rule*: the probability for an ant to move from city  $i$  to city  $j$  depends on a random variable  $q$  uniformly distributed over  $[0, 1]$  and a parameter  $q_0$ ; if  $q \leq q_0$ , then among the feasible components, the component that maximizes the product is chosen.

#### 4.3. MAX-MIN ANT SYSTEM (MMAS)

Another version of Ant System is Max-Min ant system proposed by Stützle and Hoos [15], which provide effectively, exploit good solutions. It differs from AS in that i) only the best ant adds pheromone trails and ii) the maximum and minimum values of the pheromone ( $\tau_{\max}$ ,  $\tau_{\min}$ ) are explicitly limited. All edges are initialized to  $\tau_{\max}$  and reinitialized to  $\tau_{\max}$  when nearing stagnation (idle state). It can be applied to identify potential solutions in the search space using pheromone trail and create new promising initial solution for local search. The advantage has inspired us to start a further investigation on applying MMAS for solving CO problems.

#### 4.4. RANK-BASED ANT SYSTEM ( $AS_{\text{RANK}}$ )

$AS_{\text{rank}}$  is another enhancement over AS proposed by Bullnheimer et al. [16]. In  $AS_{\text{rank}}$ , all solutions are ranked according to their fitness. The amount of pheromone deposited is then weighted for each solution, such that the more optimal solutions deposit more pheromone than the less optimal solutions.

#### 4.5. RECURSIVE ANT COLONY OPTIMIZATION

It is a recursive form of Ant System [17] which divides the whole search domain into several subdomains and solves the objective on these subdomains. The results from all the subdomains are compared and the best few of them are selected and promoted for the next level. The subdomains corresponding to the selection results are further subdivided and the process is repeated till a result of desired precision is obtained. This method has been tested on ill-posed geophysical inversion problems and work well.

#### 4.6. ANTNET

Another improved version of ACO algorithm is AntNet [18] which is developed by Di Caro & Dorigo according to the principles of ACO. It is also one of the most successful ACO-based routing protocols based on distance. In AntNet, the concepts of the forward ants and the backward ants are presented. The forward ants choose the next hop randomly according to the heuristic information values in the routing tables. All the forward ants are converted to the backward ants as soon as they arrive at the final destination. The backward ant travels back to the source through the reverse route and releases pheromones on each link passed by.

## 5. APPLICATIONS OF ACO AND CURRENT TRENDS

The initial applications of ACO were in the domain of NP-hard combinatorial optimization problems. Regarding applications, the use of ACO for the solution of scheduling problem (job, project), assignment problem (quadratic, frequency, graph coloring), networking (optical network routing, connection oriented routing), applied in fuzzy systems, knapsack problem and set covering problems. Nevertheless, the majority of the applications as well as the most successful ones belong to the class of ACO algorithms. ACO algorithms are currently among the state-of-the-art methods for solving different problems are listed in Table 1.

Table 1. Applications of Ant Colony Optimization

| Problem Name                        | Authors  | Algorithm Name     |
|-------------------------------------|--|--------------------|
| Travelling Salesman Problem         | Dorigo, Maniezzo and Colorni<br>Dorigo and Gambardella | AS<br>ACS          |
| Quadratic Assignment Problem        | Maniezzo<br>Maniezzo and Colorni                       | ANTS-QAP<br>AS-QAP |
| Frequency Assignment Problem        | Maniezzo and Carbonaro                                 | ANTS-FAP           |
| Optical Network Routing             | Navarro Varela and Sinclair                            | ACO-VWP            |
| Vehicle Routing Problem             | Gambardella, Taillard and Agazzi                       | HAS-VRP            |
| Graph Coloring                      | Costa and Hertz  | ANTCOL             |
| Sequential Ordering                 | Gambardella and Dorigo                                 | HAS-SOP            |
| Connection-Oriented Network Routing | Di Caro and Dorigo                                     | AntNet-FS          |
| Connection-less Network Routing     | Di Caro and Dorigo                                     | AntNet & AntNet-FA |

## 6. CONCLUSION AND FUTURE DIRECTION

Being a successful metaheuristic, ACO has been constructed to find solutions of NP-hard problems. The ACO has been successfully applied in various application domains such as travelling salesman problem, quadratic assignment problems, scheduling problems or graphic coloring is tackled. As such, there is no guarantee to find optimal solution always. Moreover, the obtained results may be optimal or approximate that depends on the degree of fitness of the algorithm. Therefore, it is crucial to find analysis a given task and to properly select the parameters to be executed and to perform their optimization. Recently, researchers have been dealing with the relation of ACO algorithms to other probabilistic methods for reinforcement learning and optimization.

### Reference

- [1] Dorigo M., (1992) "Optimization, Learning and Natural Algorithms", PhD thesis [in Italian], Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.
- [2] Dorigo M., and Blum C., (2005) "Ant Colony Optimization Theory: A Survey", Theoretical Computer Science, Vol. 344, No. 2-3, pp.243-278.
- [3] Christian Blum, (2005) "Ant colony optimization: Introduction and recent trends", Physics of Life Reviews, Vol. 2, No. 4, pp. 353-373.
- [4] Balasubramaniam S et. Al, (2006) "Biologically Inspired Self-Governance and Self-Organization for Autonomic Networks", IEEE.
- [5] Jesus Millor, Jose Halloy, Jean-Marc Ame and Jean-Louis Deneubourg, (2006) "Individual Discrimination Capability and Collective Choice in Social Insects", Ant colony optimization & swarm intelligence, Proceedings of 5<sup>th</sup> International Workshop ANTS 2006, Springer, LNCS 4150, pp.167-178.



- [6] Neumann F., and Witt C., (2010) "Bioinspired computation in Combinatorial Optimization - Algorithms and their Computational Complexity", Natural Computing Series, Springer, ISBN 978-3-642-16543-6.
- [7] Dorigo M., Manieo V., and Colorni A., (1991) "Positive Feedback as a Search Strategy", Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy.
- [8] Hadeli, Valckenaers P., Kollingbaum M., and Van Brussel H., (2004) "Multi-agent coordination and control using stigmergy", Computers in Industry, Vol. 53, pp.75-96.
- [9] Bonabeau E., Dorigo M., and Theraulaz G., "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, New York.
- [10] Mandeep Kaur Bedi, and Sheena Singh, (2013) "Comparative Study of Two Natural Phenomena Based Optimization Techniques", International Journal of Scientific & Engineering Research, Vol. 4, No. 3, ISSN 2229-5518.
- [11] [http://www.scholarpedia.org/article/Ant\\_colony\\_optimization](http://www.scholarpedia.org/article/Ant_colony_optimization)
- [12] Dorigo M., Maniezzo V. and Colorni A., (1996) "Ant System: Optimization by a colony of cooperating agents", IEEE Transaction SystemS, Man, Cybernet - Part B, Vol. 26, No. 1, pp.29-41.
- [13] Sorin C., Negulescu, Constantin Oprean, Claudiu V., Kifor, and Ilie Carabulea., ( 2008) "Elitist Ant System for Route Allocation Problem", 8<sup>th</sup> WSEAS International Conference on Applied Informatics and Communications (AIC'08), Greece, pp.62-67.
- [14] Dorigo M., and Gambardella L.M., (1997) "Ant Colony System: A cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pp.53-56.
- [15] Stützle T., and Hoos H.H., (2000) "MAX-MIN Ant System", Future Generation Computer System, Vol. 16, No. 8, pp.889-914.
- [16] Bullnheimer B., Hartl R.F., and Strauss C., (1997) "A New Rank Based Version of the Ant System: A Computational Study", Working Paper No.1, SFB Adaptive Information Systems and Modelling in Economics and Management Science, Vienna, Vol. 7, No.1, pp.25-38.
- [17] Nitish Sabharwal, Harshit Sharma, (2011) "A Recursive Ant Colony System Algorithm for the TSP", International Conference on Advancements in Information Technology with Workshop of ICBMG 2011, IPCSIT, Vol. 20, pp.60-66.
- [18] Di Caro G., and Dorigo M., (1998) "AntNet: Distributed Stigmergetic Control for Communications Networks", Journal of Artificial Intelligence Research, Vol. 9, No. 3, pp.317-365.