# Review: Applications of Ant Colony Optimization

*K. R. Hole[1], R. A. Meshram[2], P. P. Deshmukh[3]*

[1]PRMIT&R, Badnera, India
Department of Computer Science and Engineering
*komal9112@gmail.com*

[2]PRMIT&R, Badnera, India
Department of Computer Science and Engineering
*rupalimeshram235@gmail.com*

[3]PRMIT&R, Badnera, India
Department of Computer Science and Engineering
*pranita.33deshmukh@gmail.com*

**Abstract:** *Ant Colony Optimization (ACO) studies artificial systems that take inspiration from the behavior of real ant colonies and which are used to solve discrete optimization problems. Ants don't use vision as they are Almost blind. Shortest path is discovered via pheromone trails, which is the chemical left on the ground which is use as a signal for other ants. More pheromone on path increases probability of path being followed. When ants want to reach to the food from home they wander randomly through different paths and when any one ant finds a short path from colony to food source, other ants are more likely to follow that short path, and such positive feedback eventually leaves all ants following same path. They are Incapable of achieving complex tasks alone and rely on the phenomena of swarm intelligence for survival. ACO originally applied to Traveling Salesman Problem and Applied later to various hard optimization problems and also use for Solving NP-hard combinatorial problems.*

**Keywords:** ACO, Travelling Salesman Problem, PSP, swarm intelligence

## 1. Introduction

The ant colony algorithm is an algorithm for finding optimal paths that is based on the behaviour of ants searching for food. At first, the ants wander randomly. When an ant finds a source of food, it walks back to the colony leaving "markers" (pheromones) that show the path has food. When other ants come across the markers, they are likely to follow the path with a certain probability. If they do, they then populate the path with their own markers as they bring the food back. As more ants find the path, it gets stronger until there are a couple streams of ants travelling to various food sources near the colony.

Because the ants drop pheromones every time they bring food, shorter paths are more likely to be stronger, hence optimizing the "solution." In the meantime, some ants are still randomly scouting for closer food sources. A similar approach can be used find near-optimal solution to the traveling salesman problem. Once the food source is depleted, the route is no longer populated with pheromones and slowly decays. Because the ant-colony works on a very dynamic system, the ant colony algorithm works very well in graphs with changing topologies. Examples of such systems include computer networks, and artificial intelligence simulations of workers. ACO inspired by the foraging behavior of real ant was first introduced by Dorigo and his colleagues ([1], [2], [5]) in early 1990s and has become one of the most efficient algorithms for TSP.

Ant colony optimization (ACO) is a population based meta-heuristic that can be used to find approximate solutions to difficult optimization problems. In ACO, a set of software agents called artificial ants search for good solutions to a given optimization problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph.



**Figure 1:** Ant colony optimization

In figure 1, ants are moving from their nest to destination for food in two different ways. It walks back to the nest leaving pheromones that show the path, remaining ants will follow the same path with the help of maximum thickness of pheromones. The artificial ants (hereafter ants) incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model, that is, a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants. The easiest way to understand how ant colony optimization works is by means of an example. We consider its application to the

travelling salesman problem (TSP). In the TSP a set of locations (e.g. cities) and the distances between them are given. The problem consists of finding a closed tour of minimal length that visits each city once and only once.

## 2. Applications of ACO and current trends

The initial applications of ACO were in the domain of NP-hard combinatorial optimization problems. The largest body of ACO research is still, not surprisingly, to be found in this area. Another application that was considered early in the history of ACO is routing in telecommunication networks. The development of theoretical foundation was started by Gutjahr, who was the first to prove convergence in probability of an ACO algorithm. Concerning applications, the use of ACO for the solution of dynamic, multi objective, stochastic, continuous and mixed-variable optimization problems is a current discussion topic, as well as the creation of parallel implementations capable of taking advantage of the new available parallel hardware. Ant colony optimization is an algorithm developed by Dorigo in 1994 inspired upon stigmergic communication to find the shortest path in network. Typical examples are telephone, internet and any problem that can be described as TSP.

**Table 1:** Applications of ACO

| SN. | Problem Type | Problem Name |
|---|---|---|
| 1. | Routing | Traveling salesman (TSP) |
| | | Vehicle routing |
| | | Sequential ordering |
| 2. | Assignment | Quadratic assignment |
| | | Course timetabling |
| | | Graph coloring |
| | | Generalized Assignment |
| | | Frequency Assignment |
| 3. | Scheduling | Project scheduling |
| | | Car sequencing |
| | | Group Shop Scheduling |
| 4. | Subset | Set covering |
| | | Multiple knapsack |
| | | Maximum clique |
| | | Maximum Independent Set |
| | | Redundancy Allocation |
| 5. | Machine learning | Classification rules |
| | | Bayesian networks |
| | | Neural networks |
| 6. | Bioinformatics | Protein folding |
| | | DNA Sequencing |
| 7. | Model Problem | Network Model Problem |

### 2.1 Routing Problem

When we have various different paths to reach from source to destination and we try to choose the path which gives the shortest distance path in minimum time. To find out that shortest path from maximum available paths, these are following methods.

### 2.1.1 Traveling salesman (TSP)

The traveling salesman problem is a problem in graph theory requiring the most efficient (i.e., least total distance) Hamiltonian cycle a salesman can take through each of $n$ cities. No general method of solution is known, and the problem is NP-hard [4].

The first ACO algorithm was called the Ant system and it was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities. At each stage, the ant chooses to move from one city to another according to some rules:

1. It must visit each city exactly once;
2. A distant city has less chance of being chosen (the visibility);
3. The more intense the pheromone trail laid out on an edge between two cities, the greater the probability that that edge will be chosen;
4. Having completed its journey, the ant deposits more pheromones on all edges it traversed, if the journey is short;
5. After each iteration, trails of pheromones evaporate.

To apply ACO to the TSP, we consider the graph defined by associating the set of cities with the set of vertices of the graph. This graph is called construction graph. Since in the TSP it is possible to move from any given city to any other city, the construction graph is fully connected and the number of vertices is equal to the number of cities. We set the lengths of the edges between the vertices to be proportional to the distances between the cities represented by these vertices and we associate pheromone values and heuristic values with the edges of the graph. Pheromone values are modified at runtime and represent the cumulated experience of the ant colony, while heuristic values are problem dependent values that, in the case of the TSP, are set to be the inverse of the lengths of the edges.

### 2.1.2 Vehicle routing

The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem which asks "What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers". It generalizes the well-known Travelling Salesman Problem (TSP). Ant colony algorithms are known to have a significant ability of finding high-quality solutions in a reasonable time [9]. Often, the context is that of delivering goods located at a central depot to customers who have placed orders for such goods. The objective of the VRP is to minimize the total route cost. In 1964, Clarke and Wright improved on Dantzig and Ramser's approach using an effective greedy approach called the savings algorithm.

Main components:
- Road network
- Customers

- Vehicles
- Depots
- Drivers

Operational constraints:
- global
- for single routes
- Optimization objectives

## 2.2 Assignment problem

The task in assignment problems is to assign a set of items (objects, activities, etc.) to a given number of resources (locations, agents, etc.) subject to some constraints. Assignments can, in general, be represented as a mapping from a set I to a set J , and the objective function to minimize is a function of the assignments done. In assignment problem we have to solve any given problem with minimum resources. There are different types of assignment problem like Quadratic assignment, Course timetabling and Graph coloring problems. This problem can be solved by Ant Colony Optimization algorithm.

### 2.2.1 Graph coloring

When someone asked to help out a cartographer or a map-maker with map-coloring problem, it wants to color the countries on a map. It doesn't matter which color a country is assigned, as long as its color is different from that of all bordering countries. If two countries meet only at a single point, they do not count as sharing a border and hence can be made the same color. The cartographer is poor and can't afford many crayons, so the idea is to use as few colors as possible. Most of the algorithms used to solve the coloring problem are summarized in Hurley [7], and include sequential assignment methods, simulated annealing and genetic algorithms.

In 1852 Francis Guthrie, while trying to color the map of countries of England, noticed that four colors sufficed. Subsequently, he conjectured that 4 colors are enough to color any map. Successive efforts made to prove Guthrie's 4-color conjecture led to the development of much of graph theory. Some of which are connected by lines called edges. A vertex in a graph models some physical entity or abstract concept. An edge, which joins exactly two vertices, represents the relationship or association between the respective entities or concepts. The map of countries mentioned above can, for instance, be converted into an equivalent graph by letting each country be a vertex and connecting two vertices by an edge if the corresponding countries share a border, where sharing a border is as species above. We introduce the graph coloring problem, a classical problem in graph theory, using this informal definition of graph. In its simplest form, the graph coloring problem is to assign labels (called colors) to the vertices of a graph in such a way that no two vertices connected by an edge share the same label (color). The objective is to use the fewest number of colors possible. The graph coloring problem has a central role in computer science. It Models many significant real-world problems, or arises as part of a solution for other important problems.
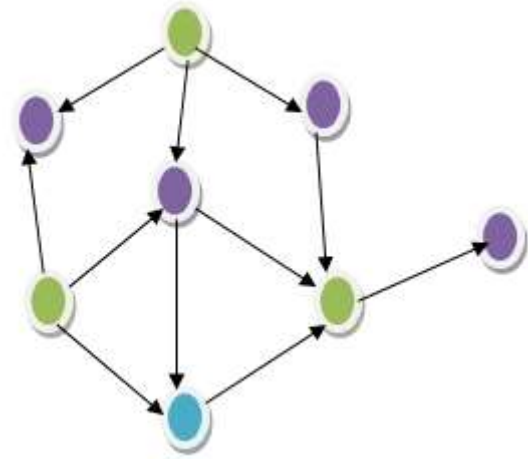
**Figure 2:** Graph Coloring Problem

In Figure 2 there are 8 nodes and used maximum 3 colors to color the nodes with all concurrent nodes contains different colors. Another ant coloring method was recently proposed in [6], where each ant is a local search instead of a constructive heuristic. The authors consider (not necessarily legal) k-colorings and try to minimize the number of conflicts. The graph coloring problem A vertex coloring of a graph, or simply Coloring for short, is an assignment of colors to the vertices such that no two adjacent vertices are assigned the same color. Alternatively, a coloring is a partition of the vertex set into a collection of vertex-disjoint independent sets. Each independent set in such a partition is called a color class. The graph coloring problem is to find a vertex coloring for a graph using the minimum number of colors possible. A X-coloring of a graph G is a coloring of G using X colors. The minimum possible value for X is called the chromatic number of G. denoted as X(G)A coloring with the fewest possible number of colors

## 2.3 Scheduling problem

The problem of determining whether a set of periodic tasks can be assigned to a set of heterogeneous processors in such a way that all timing constraints are met has been shown, in general, to be NP-hard.

### 2.3.1 Project scheduling

Project scheduling problem (PSP) is one of the important and challenging problems faced by the software project managers in the highly competitive software industry. As the problem is becoming an NP-hard problem with the increasing numbers of employees and tasks, only a few algorithms exist and the performance is still not satisfying. To design an effective algorithm for SPSP, this paper proposes an ant colony optimization (ACO) approach which is called ACS-SPSP algorithm [3]. Since a task in software projects involves several employees by splitting tasks and distributing dedications of employees to task nodes we get the construction graph for ACO. Six domain-based heuristics are designed to consider the factors of task efforts, allocated dedications of employees and task importance. Among these heuristic strategies, the heuristic of allocated dedications of employees to other tasks performs well.

## 2.4 Machine learning

Machine learning is a type of artificial intelligence that provides computers with the ability to learn without being

explicitly programmed. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.

### 2.4.1 Bayesian networks

Bayesian networks give a graphical representation of knowledge in probabilistic domains, which is becoming increasingly important in artificial intelligence by Pearl, 1998 and Jensen, 2001. A Bayesian network is a directed, acyclic graph, where the set of nodes N is in one-to-one correspondence to a set of random variables X and the set of arcs A represents direct dependence relationships between the variables. Learning Bayesian networks based on a recently introduced meta-heuristic, which has been successfully applied to solve a variety of combinatorial optimization problems: ant colony optimization (ACO). Bayesian networks are knowledge representation tools that model the independency relationships among variables for probabilistic reasoning. Classification with Bayesian networks aims to compute the class with the highest probability given a case. This special kind is referred to as Bayesian network classifiers. Since learning the Bayesian network structure from a dataset can be viewed as an optimization problem, heuristic search algorithms may be applied to build high-quality networks in medium- or large-scale problems, as exhaustive search is often feasible only for small problems[8].

## 3. Advantages and disadvantages

### 3.1 Advantages

a) They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time.

b) Algorithms found good solutions on small problems.

c) Ant Colony Optimization has been applied to a wide variety of applications

### 3.1 Disadvantages

a) Dynamic rerouting through shortest path if one node broken.

b) On static problems like TSP hard to beat specialist algorithm.

## 4. Conclusion

ACO is a relatively new meta-heuristic approach for solving hard combinatorial optimization problems. Artificial ants implement a randomized construction heuristic which makes probabilistic decisions. The cumulated search experience is taken into account by the adaptation of the pheromone trail. ACO shows great performance with the "ill-structured" problems like network routing. In ACO local search is important to obtain good results.

## References

[1] A. Bonnaccorsi, "On the Relationship between Firm Size and Export Intensity," Journal of International Business Studies, XXIII (4), pp. 605-635, 1992. (journal style)

[2] K. Deb, C. 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.

[3] A. K.N.Vitekar, S.A.Dhanawe, D.B.Hanchate, "Review of Solving Software Project Scheduling Problem with Ant Colony Optimization," Journal International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 4, pp. 1177-1182, April 2013.

[4] H.H. Crokell, Garey M, Johnson D. Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman; 1979.

[5] Dorigo, M. Maniezzo, V., and Colorni A. 1991. Positive feedback as a search strategy, Technical Report 91-016, Politecnico di Milano, Dipartimento di Elettronica, Italy.

[6] Asa Shawe-Taylor, J., Zerovnik, J. 2002. Ants and Graph Coloring, Proceedings of ICANNGA'01: 593-597.

[7] Hurley S., Thiel S.U. and Smith D.H., Frequency Assignment Algorithms, Final Report Year2 1996/97, http://www.cs.cf.ac.uk/User/SteveHurley/Ra/year2

[8] Salama, Khalid M. and Freitas, Alex A. (2013) Learning Bayesian network classifiers using ant colony optimization. Swarm Intelligence, 7 (2-3). pp. 229-254. ISSN 1935-3812.

[9] Marco Dorigo and Thomas Stutzle. Ant Colony Optimization. The MIT Press, 2004.

## Author Profile

**K. R. Hole** has received her M.E. degree in Computer Science & Engineering from Sipna College of Engineering & Technology, Amravati, India. She has published five papers in international journals. Her research interests include Artificial Intelligence and Algorithm. Currently she is working as an Assistant Professor at Prof. Ram Meghe Institute of Technology & Research, Badnera, Maharashtra, India.

**R. A. Meshram** has received her M.Tech. degree in Computer Science & Engineering from Govt. College of Engineering, Amravati, India. She has published one paper in national level and three papers in international journals. Her research interests include data mining, pattern classification. Currently she is working as an Assistant Professor at Prof. Ram Meghe Institute of Technology & Research, Badnera, Maharashtra, India.

Meghe Institute of Technology & Research, Badnera, Maharashtra, India.

**P.P. Deshmukh** has received her ME. Degree in Computer Science & Engineering from P.R.M.C.E.A.M, Badnera, Amravati, India. Currently she is working as an Assistant Professor at Prof. Ram