# Ant Colony Optimization: An Overview 

Sonika, Pardeep Kumar Mittal<br>Department of Computer Science and Applications<br>Kurukshetra University, Kurukshetra


#### Abstract

Ant colony optimization is a met heuristic approach belonging to the model based search algorithm. It is a paradigm for designing met heuristic algorithm for combinatorial problem In this paper we discuss the Ant colony system. Ant colony system is one of the best algorithm of ant colony optimization. First we discuss the optimization, and one of the optimization problem is combinatorial problem. To solve these types of problem Ant colony optimization was proposed. This paper contains description about Ant system and algorithm. Finally we discuss various applications of Ant colony optimization.


## Keywords

Optimization,combinatorial optimization,Ant colony optimizationMetaheuristic Algorithm,Ant system

## 1. Introduction

Optimization an act, process, or methodology of making something (as a design, system, or decision) as fully perfect, functional, or effective as possible. It refers to choosing the best element from some set of available alternatives. More generally, it means finding "best available" values of some objective function given a defined domain, including a variety of different types of objective functions and different types of domains.

Optimization problem a computational problem in which the object is to find the best of all possible solution[11].Like ,the traveling salesman problem is an optimization problem, while the corresponding decision problem asks if there is a Hamiltonian cycle with a cost less than some fixed amount k.

An optimization problem can be represented in the following way: Given: a function $\mathrm{f}: \mathrm{A} \longrightarrow \mathbf{R}$ from some set A to the real numbers

Sought: an element $x 0$ in A such that $f(x 0) \leq f(x)$ for all x in A ("minimization") or such that $\mathrm{f}(\mathrm{x} 0) \geq \mathrm{f}(\mathrm{x})$ for all x in A ("maximization").

There are different types of optimization problem like Infinite-dimensional optimization, Robust programming, Nonlinear programming etc. One of the problem is combinatorial problem.Combinatorial optimization is concerned with problems where the set of feasible solutions is discrete or can be reduced to a discrete one.
Combinatorial optimization is a topic that consists of finding an optimal object from a finite set of objects. In many such problems, exhaustive search is not feasible. It operates on the domain of those optimization problems, in which the set of feasible solutions is discrete or can be reduced to discrete, and in which the goal is to find the best solution. Some common problems involving combinatorial optimization are the traveling salesman problem and the minimum spanning tree problem.Combinatorial optimization is a subset of optimization that is related to operations research algorithm theory, and computational complexity
theory. It has important applications in several fields, including artificial intelligence, machine learning, mathematics, and software engineering. There are many algorithm and techniques to solve combinatorial problem but one of the most prominent technique is ant colony optimization. This new heuristic has the following desirable characteristics:

1) It is versatile. in that it can be applied to similar versions of the same problem; for example, there is a straightforward extension from the travelling salesman problem(TSP) to the asymmetric travelling salesman problem(ATSP).
2) It is robust. it can be applied with only minimal changes to other combinatorial problems such as the quadratic assignment problem(QAP) and the job shop scheduling problem(JSP).
3) It is a population based approach. This is interesting because it allows the exploitation of positive feedback as a search mechanism.

## 2. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a prototype for designing meta-heuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms [1,13]is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.
Meta-heuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. The meta-heuristic part permits the low-level heuristic to obtain solutions better than those it could have achieved alone, even if iterated.
ACO $[1,10]$ is a class of algorithms, whose first member, called Ant System, was initially proposed by Colorni, Dorigo and Maniezzo [13,10,12]. The main under-lying idea, loosely inspired by the behavior of real ants, is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. The collective behavior emerging from the interaction of the different search threads has proved effective in solving combinatorial optimization (CO) problems[5].A combinatorial optimization problem is a problem defined over a set $C=c_{1}, \ldots \ldots \ldots \ldots c_{n}$ basic components.A set of computational concurrent and asynchronous agents (a colony
of ants) moves through states of the problem corresponding to partial solutions of the problem to solve. They move by applying a stochastic local decision policy based on two parameters, called trails and attractiveness. By moving, each ant incrementally constructs a solution to the problem. When an ant completes a solution, or during the construction phase, the ant evaluates the solution and modifies the trail value on the components used in its solution. This pheromone information will direct the search of the future ants.

An ACO algorithm includes two more mechanisms: trail evaporation and, optionally, daemon actions. Trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component. Daemon actions can be used to implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective[8].
An ant is a simple computational agent, which iteratively constructs a solution for the instance to solve. Partial problem solutions are seen as states. At the core of the ACO algorithm lies a loop, where at each iteration, each ant moves (performs a step) from a state $\tau$ to another one $\Psi$,corresponding to a more complete partial solution. That is, at each step $\sigma$. each ant k computes a set $A_{k}^{\sigma(\tau)}$ of feasible expansions to its current state, and moves to one of these in k probability. The probability distribution is specified as follows. For ant k, the probability $\rho_{\tau \Psi}^{K}$ of moving from state $\tau$ to state $\Psi$ depends on the combination of two values:
1.The attractiveness $\eta_{\tau} \Psi$ of the move as computed by some heuristic indicating a posteriori indication of the desirability of the move.
2. Trails level $\tau_{\tau} \Psi$ indicating of the move how proficient it has been in the past to make that particular move: it represents therefore an a posteriori indication of the desirability of that move.

Trails are updated usually when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions, respectively.

## 3. ANT SYSTEM

The importance of the original Ant System (AS)[13,7,18] resides mainly in being the prototype of a number of ant algorithms which collectively implement the ACO paradigm. The move probability distribution defines probabilities $p^{t \Psi}$ to be equal to 0 for all moves which are infeasible (i.e., they are in the tabu list of ant k , that is a list containing all moves which are infeasible for ants k starting from state $\tau$ ), otherwise they are computed by means of formula (1.1), where a and b are user-defined parameters $\quad(0 \leq \alpha, \beta \leq 1)$, tabu is the tabu list of ant k , while parameters $\alpha$ and $\beta$ specify

$$
\mathrm{p}_{1 \psi}^{\mathrm{k}}=\left\{\begin{array}{cl}
\frac{\tau_{1 \psi}^{\alpha}+\eta_{1 \psi}^{\beta}}{\sum_{(1 \zeta) \notin t a b u_{k}}\left(\tau_{1 \zeta}^{\alpha}+\eta_{1 \zeta}^{\beta}\right)} & \text { if }(1 \psi) \notin \text { tabu } \mathrm{k}_{\mathrm{k}} \\
0 & \text { otherwise }
\end{array}\right.
$$

(1.1)In formula 1.1 , tabu is the tabu list of ant k , while parameters $\alpha$ and $\beta$ specify the impact of trail and attractiveness, respectively. After each iteration $t$ of the algorithm, i.e., when all ants have completed a solution, trails are updated by means of formula (1.2):

$$
\begin{equation*}
\tau_{t \Psi}(\tau)=\rho \cdot \tau_{t \Psi}(\tau-1)+\Delta \tau_{t \Psi} \tag{1.2}
\end{equation*}
$$

where $\Delta \tau_{t \Psi}$ represents the sum of the contributions of all ants that used move to $(\tau \Psi)$ construct their solution, $\rho, 0 \leq \rho \leq \mathbf{1}$ is a user-defined parameter called evaporation coefficient, and $\boldsymbol{\Delta} \tau_{t \Psi}$ represents the sum of the contributions of all ants that used move $(\tau \Psi)$ to construct their solution. The ants' contributions are proportional to the quality of the solutions achieved, i.e., the better solution is the higher will be the trail contributions added to the moves it used. For example, in the case of the TSP, moves correspond to arcs of the graph, thus state 1 could correspond to a path ending in node i ,the state $\Psi$ to the same path but with the arc (ij) added at the end and the move would be the traversal of $\operatorname{arc}(\mathrm{ij})$. The quality of the solution of ant k would be the length $L_{K}$ of the tour found by the ant and formula (1.3) would become

$$
\begin{gather*}
\tau_{i j}(t)=\rho \tau_{i j}(t-1)+\Delta \tau_{i j}  \tag{1.3}\\
\Delta \tau_{i j}=\sum_{k=1}^{m} \Delta \tau_{i j}^{k}
\end{gather*}
$$

with
where $m$ is the number of ants and

$$
\Delta \tau_{i j}^{k}
$$ is the amount of trail laid on edge (ij) by ant k , which can be computed as

$$
\begin{gathered}
\Delta \tau_{i j}^{k}=\frac{Q}{L_{k}} \text { if ant } k \text { uses (ij)in its tour } \\
0 \quad \text { otherwise }
\end{gathered}
$$

Q being a constant parameter.The ant system simply iterates a main loop where $m$ ants construct in parallel their solutions, thereafter updating the trail levels. The performance of the algorithm depends on the correct tuning of several parameters, namely: $\alpha, \beta$ relative importance of trail and attractiveness $\rho$ trail persistence $\tau_{i j}(0)$ initial trail level, m, number of ants, and Q , used for defining to be of high quality solutions with low cost.

## Algorithm

## 1. (Initialization)

Initialize $\tau_{t \Psi}$ and $\eta_{t \Psi}, \forall(t \Psi)$.
2. (Construction)

For each ant $k$ (currently in state t) do
Repeat
Choose in probability the state to move into.
$t a b u_{k}$.
Append the chosen move to the k-th ant's set

Until ant k completed its solution.

End for
3. (Trail update)

For each ant move $t \Psi$ do
Compute $\Delta \tau_{t} \Psi$
Update the trail matrix
End for
4.(Terminating condition)

If $\operatorname{not}$ (end test) go to step 2 .

## 4. APPLICATION OF ANT COLONY OPTIMIZATION

Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to fold protein or routing vehicles and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. As a very good example, ant colony optimization algorithms have been used to produce near-optimal solutions to the travelling salesman problem. 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.

### 4.1 Applications to NP-Hard Problems

In the case of ACO, the research initially consisted of testing the algorithms on TSP. Subsequently, other NP-hard problems were also considered[14,15]. Many of the tackled problems can be considered as falling into one of the following categories: routing problems as they arise, for example, in the distribution of goods; assignment problems, where a set of items (objects, activities, etc.) has to be assigned to a given number of resources (locations, agents, etc.) subject to some constraints;scheduling problems, which-in the widest senseare concerned with the allocation of scarce resources to tasks over time; and subset problems, where a solution to a problem is considered to be a selection of a subset of available items.

### 4.2 Applications to telecommunication networks

ACO algorithms have shown to be a very effective approach for routing problems in telecommunication networks where the properties of the system, such as the cost of using links or the availability of nodes, varies over time. ACO algorithms were first applied to routing problems in circuit switched networks (like telephone networks) and then in packetswitched networks (like local area networks or the Internet). A well known example is Ant Net. Ant Net has been extensively tested, in simulation, on different networks and under different traffic patterns, proving to be highly adaptive and robust.

### 4.3 Applications to industrial problem

The success on academic problems has raised the attention of a number of companies that have started to use ACO algorithms for real-world applications. Among the first to exploit algorithms based on the ACO met heuristic is EuroBios (www.eurobios.com). They have applied ACO to a number of different scheduling problems such as a continuous two-stage flow shop problem with finite reservoirs. The problems modeled included various real-world constraints such as setup times, capacity restrictions, resource compatibilities and maintenance calendars.

## 5. CONCLUSIONS

Ant Colony Optimization has been and continues to be a fruitful paradigm for designing effective combinatorial optimization solution algorithms. After more than ten years of studies, both its application effectiveness and its theoretical groundings have been demonstrated, making ACO one of the most successful paradigm in the met heuristic area. This overview tries to propose the reader both introductory elements and pointers to recent results, obtained in the different directions pursued by current research on ACO.As the ACO research field is currently flourishing, we expect to see many of these problems solved in near future. As a final comment, we note that the ACO research is not only about theory. No doubt new results will both improve those outlined here and widen the area of applicability of the ACO paradigm.

## 6. REFERENCES

[1] 1]M.Dorigo,Ant colony optimization web page,http://iridia.ulb.ac.be/mdorigo/ACO/ACO.html
[2] [2] M. E. Bergen, Canstraint-based assembly line sequencing, Lecture Notes in Computer Science, 2001
[3] [3]L. Bianchi, L.M. Gambardella, M.Dorigo. An ant colony optimization approach to the probabilistic traveling salesman problem. In Proceedings of PPSN-VII, Seventh Inter-national Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, 2002
[4] [4] E. Bonabeau, M. Dorigo, G. Theraulaz, Nature, Volume 406, Number 6791, Pag. 39-42 (2000)
[5] [5]B. Bullnheimer, R.F. Hartl, and C. Strauss, A new rank-based version of the ant system: a computational study, Central European Journal of Operations Research 7 (1)(1999), 25-38.
[6] [6]T. Bäck and H.-P. Schwefel, An overview of evolutionary algorithms for parameter optimization, Evolutionary Computation 1(1), (1993), 1-23.
[7] [7] M. den Besten, T. Stützle, M. Dorigo, Ant colony optimization for the total weighted tardiness problem, Parallel Problem Solving from Nature: 6th international conference, September 2000. Springer Verlag.
[8] [8] A. Colorni, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies, Proceedings of ECAL'91, European Conference on Artificial Life, Elsevier Publishing, Amsterdam, 1991.
[9] [9] C. Chao-Hsien, G. JunHua, H. Xiang Dan, G. Qijun, A heuristic ant algorithm for solving QoS multicast routing problem, in Proceedings of the 2002 congress on Evolutionary Computation, Honolulu, USA
[10] [10] M. Dorigo, Optimization, learning and natural algorithms, Ph.D. Thesis, Politecnico di Milano, Milano, 1992.
[11] [11] X Hu, J Zhang, and Y Li (2008). Orthogonal methods based ant colony search for solving continuous optimization problems. Journal of Computer Science and Technology, 23(1), pp.2-18.
[12] [12] M. Dorigo, M. Birattari \& T. Stützle, 2006 Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. TR/IRIDIA/2006-023
[13] [13] R.Schoonderwoerd, O.Holland, J.Bruten, and L.Rothkrantz, ìAnt-based load balancing
telecommunications networks,î Adaptive Behavior, vol. 5, no. 2, pp. 169ñ207, 1996.
[14] [14] G.Di Caroand M.Dorigo, ìAntNet: Distributed stigmergetic control for communications networks,î Journal of Artificial Intelligence Research, vol. 9, pp. 317ñ365, 1998.
[15] [15] W.J.Gutjahr, ìA converging ACO algorithm for stochastic combinatorial optimization,1̂ ofl, Eds., Berlin, Germany:in Proc. SAGA 2003, ser. LNCS, A. Albrecht and T. Steinh Springer Verlag, vol. 2827, pp. 10ñ25, 2003.

