Ant Colony Algorithms: A Brief Review and Its Implementation

Renu Jangra
Ph.D,Research Scholar
Department of Computer Science & Applications, Kurukshetra University, Kurukshetra-136119
Email : renu.jangra2010@gmail.com
Dr.Ramesh Kait
Assistant Professor
Department of Computer Science & Applications, Kurukshetra University, Kurukshetra-136119
Email : ramesh.kait@kuk.ac.in

Abstract

Ant Colony optimization is a method to find the shortest possible path from source to sink. ACO algorithms are inspired by the social behavior of ants that use stigmergic behavior to search the food. A chemical substance left by ants, called pheromone , help them to find the food .In this paper, we present the basic behavior of ants for searching food, preliminary study of different papers related to ACO , basic ant colony algorithms and its extensions.

Keywords: *Ant colony Optimization,ACO algorithms;*

1. Introduction:

Ant Colony Optimization is a metaheuristic method that takes inspiration from the collective behavior of real ant colony or social insects[1]. Ants behavior inspired a number of methods and techniques among which the most successful and studied is the general purpose optimization technique ant colony optimization (ACO). Foraging behavior of ant species inspired ACO. These ants deposit a chemical substance called pheromone on the ground in order to mark some positive path that should be followed by other members of the ant colony. Ant colony optimization exploit a same mechanism for solving optimization problems[2].ACO algorithm is a member of the swarm intelligence methods. It is a probabilistic technique for finding close to optimal path through a problem space.Dorigo.M, Maniezzo.V and Colomi.A; Italian scholars have through simulating the foraging behavior of ant colony & put forward a simulated evolutionary algorithm based on populations i.e Ant Colony Optimization (ACO). It is a random search algorithm which seeks the best solution through analyzing the evolution process of ant group collected of candidate solutions. This algorithm is applied in fields such as combinatorial optimization, network routing, data mining ,function optimization and robot path planning etc with good effect during past over ten years. This algorithm has good distributed calculative mechanism and strong robustness .It is easy to merge with other methods and the nice performance has been shown on resolving the complex optimization problems.ACO is more suitable for solving the optimization problems in complex environments. Therefore, it has a great academic meaning and engineering value to make theoretical analysis and applicable study to ACO. Also ,ACO has been useful in data analysis, solution of multi-robot coordination problems as well as in fields such as electric, communication, mining , chemical , water conservancy, architecture and traffic, etc[3].

By the careful observation and study, bionomists came to know that ants will leave a chemical substance called pheromone on their way to look for food. The more ants move on the same way, the more substance will be left on that path. This is done with two objectives (a) it allows ants to find their way back to the nest(b) it allows other ants to know the path they have taken, so that the others can also follow them. As a result, there will be a bigger probability for the ants to follow the same path .By this way of communication; ants find their food at last. The more time the ant takes, to travel from the nest to the food source and back to the nest, the pheromones have to evaporate ,that is, over the time, the pheromone evaporates and thus its quantity reduces [4].

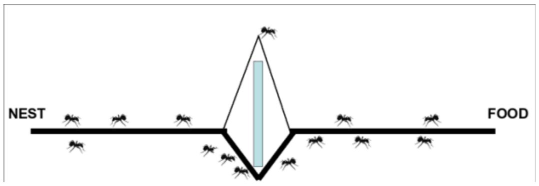

Fig 1. Ant Colony Optimization

All solutions have the same amount of information at the initial stage . As we move on in the algorithm, the amount of information on the better solution will increase and the algorithm gradually becomes convergent. Ant Colony Optimization[5] is always the highlight of key algorithm. There are many improvements about the ACO algorithm such as: the improvements of algorithm in self-adaptive, the improvements of increasing the diversity of various group, the improvements of enhancing local search,combining with the global optimization algorithm and combining with deterministic local optimization algorithm etc.

Basic Ant Colony Algorithm and Flowchart

Following is the pseudo-code of the ACO metaheuristic procedure:

```
procedure ACOMetaheuristic
   ScheduleActivities
      ConstructAntsSolutions
      UpdatePheromones
      DaemonActions
//optional
  end-ScheduleActivities
end-procedure
```

ACO algorithm mainly contains the three procedures: ConstructAntsSolutions; UpdatePheromones and DaemonActions.

ConstructAntsSolutions: It manages the group of ants moving to the different neighbor nodes of the problem's construction graph. During the move, ants use both pheromone trail and heuristic information to build the solution of the optimization problem. Once an ant has built a solution or partial solution that will be used by the UpdatePheromones procedure to decide how much pheromone to deposit [6].

UpdatePheromones: Pheromone trails are modified in this process. The trail value of that path will increase on which the pheromone is leave by the ant otherwise pheromone evaporation contributes to decrease the trail value. The addition of new pheromone increases the probability that, the same path will be used by the other ants and that results in a good solution.

DaemonActions: The actions which cannot be performed by a single ant because they require access to nonlocal information—interact in the solution process[7].

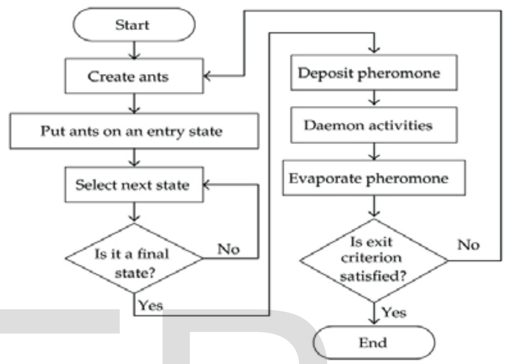

Fig 2. Flowchart of ACO

In the flowchart, initially ants are created and placed at their position(state).If ants does not reach their final state then they move further and choose the next state. If they reached the final state , then pheromone is updated and corresponding daemon actions are constructed. Next check the exit criteria, if satisfied then exit otherwise again start from the first , that is, ants are created.

2. Preliminary Study:

Oscar Cordan, Francisco Herrera, Thomas Stutzle [1] generates a review on how to apply ACO algorithms to the combinatorial problems, brief of current applications. Most of the existing approaches are explained and also some results are summarized regarding relationship between ACO and some metaheuristics.

Marco Dorigo , Mauro Birattari , and Thomas St¨utzle [2] presented introduction of ACO algorithms, application and recent hot topics of ACO. These days hundreds of researcher globally applying ACO to classic *NP*-hard optimization problems, while simply a few works concern variations that include dynamic and stochastic aspects as well as multiple objectives.

Stephen A. Adubi, Sanjay Misra [3] presented a review on different variants of ACO algorithms and also compare the result obtained by applying on different combinatorial problems. The scheduling problem is concerned to both users and grid systems. The Gang Scheduling approach achieves optimal schedule than the traditional Ant colony algorithm.

XUE Xue-dong , CHENG Xu-de , XU bing, WANG Hong-li,JIANG Cheng-peng [4] introduces the ACO algorithms and also some improvements regarding the basic AS algorithm. The current research is still in the simulation stage , the rigorous mathematical explanation and complete theoretical system have not been formally formed yet about the algorithm.

Ying Pei, Wenbo Wang, Song Zhang [5] proposed an ACO algorithm that work efficiently in multicore computing environment. This paper presents an improved ACO for multicore computing. It achieves an proficient, parallel ACO, solving the challenge which has been planned to researchers to some extent.

Denis Darquennes [6] explains the basic of ACO, its algorithms,NP – complete and TSP problems , how to construct ACO solution and their experimental results. Denis tried to define the memory in another way than those envisaged by the formal definition of Ant Colony Optimization. In the version we have developed, we try to make a difference between states that are identical from the point of view of Ant Colony Optimization, associating the memory with pairs of “sequence of components - component".

Sapna Katiyar, Ibraheem [8] , Marco Dorigo and Thomas St¨utzle [7] [10], M. Dorigo and K. Socha [15] gives their review on the ACO and its extension in growing research field.

Christian Blum [9] describes how discrete optimization can be applied on ACO ,also provide some examples on recent research and hybridization with classical techniques artificial intelligence (AI) and operation research (OR). They outlined the general framework of the ACO metaheuristic and presented some of the most successful ACO variants today. We summarized , the existing theoretical results and outlined the latest developments concerning the adaptation of ACO algorithms to continuous optimization.

Mainly ACO has problems like slow convergence and easily falling to local minimum points. Wang, Zheng and Xianmin, Ma [11] introduce the algorithm that remove the following problem by the policy of dynamic path selection and by paralleling updating the global and local information to regulate the ant search direction. So, that algorithm improve the solution and guaranteed convergence speed**.**

Shigang Cui, Shaolong Han [12] explains the fundamental principle, model, prons and cons of ant colony algorithm, the concrete realization process of ant colony algorithm is plant in solving the traveling salesman problem (TSP) and the simulation that shows, the solution is practicable.

S. D. Shtovba [13] review the basic of ACO and its application. He also explain the techniques like local search techniques, use of elitist ants , candidate lists etc. that can be used to improve the ant algorithms.

Ying Pei, Wenbo Wang, Song Zhang [14] develops an algorithm that works in multicore computing environment efficiently. The issue of the parameter optimized design in PID control, comparing the results of ACO design to the results of genetic algorithm design. The final results present that the ACO has the efficiency and application value of a new simulated evolutionary optimization method.

N.M.A AI Salami [16] describes the problem in terms of finite state automata and also the input-output specification of the problem. With the help of input-output trajectory information, program solve the problem. The results indicated that theoretical bases, can enhance efficiency and performance of automatic programming system, leading to an increase in the system productivity and letting the concentrate to be done on problem specification only.

3. ACO Algorithm and its successors:

The Ant System (AS) was the first ACO algorithm developed by the Professor Dorigo in 1992.This algorithm uses TSP as an example application. Different variants of AS takes inspiration from ant system algorithm and today, these are the most successful ACO algorithms. Difference between AS and its variants [8][9] are the way the pheromone update is done as well as the organization of the pheromone trails.

3.1 Ant System

Main phases are: Ant’s solution Construction and Pheromone update.Rough estimation of initialization of the pheromone trails is obtained by putting

$\forall$ (i,j) , $\tau_{i,j} = \tau_0 = m/C^{nn}$, where m is the number of ants and C^{nn} is the length of a tour generated by the nearest-neighbor heuristic.

Ant’s solution Construction:

Ant ‘k’ decide which city to visit next by apply the probabilistic action choice rule called random proportional rule. Currently, ant k is in city i and want to go to city j with probability given by

$$p_{i,j}^{k} = \frac{[\tau_{i,j}]^{\alpha}\,[\eta_{i,j}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{i,j}]^{\alpha}\,[\eta_{i,j}]^{\beta}} \,, \quad \text{if} \quad j \in N_i^{k}$$

where $\eta_{i,j} = 1/d_{i,j}$
$d_{i,j}$ = distance between cities i and j
$\tau_{i,j}$ = intensity of pheromone trail between cities i and j
$\eta_{i,j}$ = visibility of city j from city i
α = parameter that decide the relative control of the pheromone trail
β = parameter weighting the relative value of heuristic information $\eta_{i,j}$ in the probability function

Pheromone update:

Pheromone trails are updated , when all the ants completed their tour. Pheromone evaporation is given by

$$\tau_{i,j} \leftarrow (1-\rho)\,\tau_{i,j} \quad , \quad \text{where} \quad 0 < \rho \leqslant 1$$

ρ is used to avoid unlimited accumulation of pheromone trail.
After evaporation, all ants deposit pheromone on the path that they have crossed in their tour:

$$\tau_{i,j} \leftarrow \tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \quad , \quad \forall\,(i,j) \in L$$

where $\Delta\tau_{i,j}^{k}$ = amount of pheromone ant ‘k’ deposit on the arc, it has visited. It is defined as:

$\Delta\tau_{i,j}^{k} = 1/C^{k}$, if arc (i,j) belongs to T^{k}
otherwise zero

3.2 Elitist Ant System (EAS):

EAS is the first improvement of AS. In this algorithm[10], an additional reinforcement is provide to the arcs belonging to the best tour found while the start of the algorithm; this tour is denoted T_{bs} (best-so-far tour) in the following.
Pheromone update:

T_{bs} is achieved by adding a quantity e/C_{bs} to its arcs.

Where e is the parameter that defines the weight given to the best-so-far-tour $T_{bs.}$ C_{bs} is its length.

$$\tau_{i,j} \leftarrow \tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} + e\,\Delta\tau_{i,j}^{bs} \;,$$

where $\Delta\tau_{i,j}^{bs} = 1/C_{bs}$ if arc(i,j) belongs to T_{bs}
otherwise zero.

3.3 Rank Based Ant System(AS_{Rank}):

In this, the amount of pheromone deposit by each ant, decreases with its rank [6].Also, the largest amount of pheromone in each direction, is deposit by the best-so-far ant.
Pheromone update:

- Firstly, the ants are sorted by increasing tour length and the pheromone quantity deposit by ant is weighted according to the rank ‘r’ of the ant.
- In each iteration, only the (w-1) best ranked ants and the ant that created the best-so-far tour are allowed to deposit pheromone.
- The best-so-far tour gives the strongest feedback, with weight w, the r^{th} best ant of the current iteration contributes to pheromone updating with the value $1/C^{r}$ multiplied by a weight given by max(0, w-r). So, the ASrank pheromone updation is:

ISSN 2229-5518

$$\tau_{i,j} \leftarrow \tau_{i,j} + \sum_{r=1}^{w-1} (w-r)\, \Delta \tau_{i,j}^{r} + w\, \Delta \tau_{i,j}^{bs}$$

where $\Delta \tau_{i,j}^{r} = 1/C^{r}$ and $\Delta \tau_{i,j}^{bs} = 1/C^{bs}$

3.4 MAX-MIN Ant System (MMAS):

It contains four [11] modification w.r.t AS.

- The best-so-far ant and the ant that produced the best tour in the current iteration,is allowed to deposit pheromone.
- It limits the possible range of pheromone trail values to the interval $[\tau_{min}, \tau_{max}]$.
- The pheromone trails are initialized to the upper pheromone trail limit, which, together with a small pheromone evaporation rate, increases the exploration of tours at the start of the search.
- Pheromone trails are initialized each time the system approaches stagnation or when no improved tour has been generated for a certain number of consecutive iterations.

Pheromone update: When, all the ants constructed a tour, pheromones are updated by applying evaporation as in AS, followed by the deposit of new pheromone as below:

$$\tau_{i,j} \leftarrow \tau_{i,j} + \Delta \tau_{i,j}^{best}, \text{ where } \Delta \tau_{i,j}^{best} = 1/C^{best}$$

For best-so-far ant , $\Delta \tau_{i,j}^{best} = 1/C^{bs}$ and for iteration best $\Delta \tau_{i,j}^{best} = 1/C^{ib}$ where C^{ib} is the length of iteration best tour.

τ_{min} and τ_{max} on possible pheromone values on any arc are forced in order to avoid search stagnation and also , the forced pheromone trail limits have the outcome of limiting the probability pij of selecting a city j when an ant is in city i to the interval $[\tau_{min}, \tau_{max}]$ with $0 < p_{min} \leqslant p_{ij} \leqslant p_{max} \leqslant 1$.

3.5 Ant Colony System (ACS):

Three modification w.r.t AS.

- It exploits the search knowledge accumulated by the ants more strongly than AS does through the use of a more violent action choice rule.
- Pheromone evaporation and pheromone deposit take place only on the arcs belonging to the best-so-far tour.
- Each time an ant use an arc (i, j) to move from city i to j, it removes some pheromone from the arc to increase the searching of substitute paths.

Ant's solution Construction:

According to pseudorandom proportional rule , when located at city i , ant k moves to a city j , is given by

$j = \operatorname{argmax}_{l \in N_i^k} \{ \tau_{il} [\eta_{il}]^{\beta} \}$ if $q \leqslant q_0$, otherwise J

where , q is a random variable uniformly distributed in [0,1]
q_0 is a parameter, J is a random variable.

Pheromome Update Globally:

One ant ,the best-so-far ant,is allowed to add pheromone after each iteration.

Updation is done by:

$\tau_{ij} \leftarrow (1-\rho)\, \tau_{ij} + \rho\, \Delta \tau_{i,j}^{bs}$, for all (i,j) belongs to T^{bs} where $\Delta \tau_{i,j}^{bs} = 1/C^{bs}$

Pheromome Update Locally:

The ants use a local pheromone update rule that they apply immediately after having crossed an arc (i,j) during the tour construction:

$\tau_{ij} \leftarrow (1-\xi)\, \tau_{ij} + \xi\, \tau_0$ where $0 < \xi < 1$

4. Conclusion:

In this paper, we explain the basic principle of ant colony optimization which works on the foraging behavior of ants. After review the different papers of ACO, we conclude that the different variants of ACO algorithms are used in solving various combinatorial optimization problems Today, hundreds of researchers applying ACO to classic *NP*-hard optimization problems, while few works on concern variations .The current research is still in the simulation stage , the rigorous mathematical explanation and complete theoretical system have not been formally formed yet about the algorithm.The overview of different basic ant colony algorithms, explained in this paper , can be used in solving future research problems.

5. References:

[1] Oscar Cordan, Francisco Herrera, Thomas Stutzle ,"*A Review on the Ant Colony Optimization Metaheuristic : Basis, Models and New Trends*", Mathware and Soft Computing 9(2002)

[2] Marco Dorigo , Mauro Birattari ,and Thomas St¨utzle," *Ant Colony Optimization*", IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE | NOVEMBER 2006

[3] Stephen A. Adubi, Sanjay Misra," *A Comparative Study on the Ant Colony Optimization Algorithms*", 2014 IEEE

[4] XUE Xue-dong ,CHENG Xu-de , XU bing, WANG Hong-li,JIANG Cheng-peng," *The basic principle and application of Ant colony optimization algorithm*", 2010 IEEE

[5] Ying Pei, Wenbo Wang, Song Zhang," *Basic Ant Colony Optimization*", International Conference on Computer Science and Electronics Engineering,2012

[6] Denis Darquennes," *Implementation and Applications of Ant Colony Algorithms*", Facultes Universitaires Notre-Dame de la Paix, Namur ,Institut d'Informatique Annee academique 2004-2005

[7] Marco Dorigo and Thomas Stutzle ,"*Ant Colony Optimization*", Massachusetts Institute of Technology 2004

[8] Sapna Katiyar, Ibraheem," *Ant Colony Optimization: A Tutorial Review*", MR International Journal of Engineering and Technology, Vol. 7, No. 2, December 2015

[9] Christian Blum," *Ant colony optimization: Introduction and recent trends*", Elsevier,2005

[10] Marco Dorigo and Thomas St¨utzle," *The Ant Colony Optimization Metaheuristic:Algorithms, Applications, and Advances*".

[11] Wang, Zheng and Xianmin, Ma," *Study of Ant Colony Optimization Algorithm Based on Adaptive Theory*", International Conference on Artificial Intelligence and Computational Intelligence,2010

[12] Shigang Cui, Shaolong Han," *Ant Colony Algorithm and Its Application in Solving the Traveling Salesman Problem*", Third International Conference on Instrumentation, Measurement, Computer, Communication and Control,2013

[13] S. D. Shtovba," *Ant Algorithms: Theory and Applications*"**,** Programming and Computer Software, Vol. 31, No. 4, 2005, pp. 167–178. Translated from Programmirovanie, Vol. 31, No. 4, 2005.

[14] Ying Pei, Wenbo Wang, Song Zhang," *Basic Ant Colony Optimization*", International Conference on Computer Science and Electronics Engineering,2012

[15] M. Dorigo and K. Socha, "*An introduction to ant colony optimization*,"Handbook of Metaheuristics, Brussels : IRIDIA, 2006, vol. 26, no. l.ISSN 1781-3794.

[16] N.M.A AI Salami, "*Ant colony optimization algorithm*," UbiCC Journal, vol. 4, no. 3, pp. 823-826,2009.