International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Ant Colony Optimization: A Survey

Keisam Thoiba Meetei[1], Dr. Subhajit Ghosh[2]

[1]Galgotias University, School of Computing Science and Engineering, Plot No.2, Sector 17-A
Yamuna Expressway, Greater Noida, Gautam Buddh Nagar, Uttar Pradesh, India

[2]Professor, School of Computing Science and Engineering, Galgotias University, Plot No.2, Sector 17-A
Yamuna Expressway, Greater Noida, Gautam Buddh Nagar, Uttar Pradesh, India

Abstract: ***This paper represents simple introduction of Ant Colony Optimization and some of its main algorithms. The algorithmic concepts and mathematical equations of Ant Colony Algorithms involved in finding solutions of complex optimization problems are shown. Three of its main algorithms are discussed here. The difference among these algorithms is clearly shown by their algorithmic steps presented in this paper. Ant Colony Optimization comes under Swarm Intelligence and is considered as meta-heuristic searching algorithms. The paper shows how the Travelling Salesman Problem is solved by using these algorithms.***

Keywords: Pheromone, Probabilistic, Meta-Heuristic, Iterative, Combinatorial, Swarm, Bio-Inspired

1. Introduction

In computer science and operations research, the Ant Colony Optimization Algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm was proposed by Marco Dorigo in 1992. It is a part of Swarm Intelligence [1][2][3]. Swarm Intelligence also known as Bio-inspired computing or biologically inspired computing is a field that studies biological behavior of objects and its emergence. Biologically inspired computing is a major subset of natural computation [4][7]. The main aim of this study is to develop artificial intelligent systems that can solve complex problems and to improve the usage of computers by mimicking the natural or biological behavior of objects [5].

Ants usually wander randomly through space searching for food source. Ants will start searching from their colony moving randomly in all directions. If an ant finds food, then the ant returns to colony. While returning back towards colony the ant will leave a trail of chemical substances called pheromone along the path. By doing so other ants can detect the pheromone left by the previous ant and they follow the same path. The path is determined by the amount of concentration of pheromone along the path. It is by nature that the pheromone will start evaporating over time reducing the concentration of pheromone thus leading to poor strength of the path. Considering this condition, a shorter path will be suited for ants.

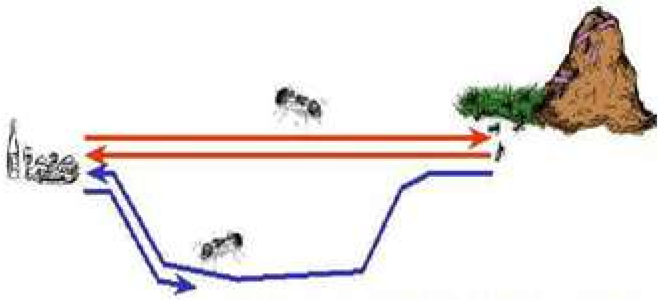

Figure 7: Ants randomly wandering for food source

Every ant will follow the shorter path thus keep adding pheromone which makes the concentration strong enough to against evaporation. This makes the emergence of shortest path from colony to food. Seeking a path between the colony and a food source is the natural behavior of ants. Based on this biological behavior, the algorithm was developed aiming to search for an optimal path in a graph [13]. The fundamental approach underlying ACO is an iterative process. ACO is a paradigm for designing Meta heuristic algorithms for combinatorial optimization problem [6].

Many ACO algorithms have been developed by various researchers. Extensions to the existing ACO algorithms have also been proposed and implemented successfully.

Table 1: List of popular Ant Colony Optimization algorithms

Year	*Algorithm*	*Authors*
1991	Ant System (AS)	Dorigo et al.
1992	Elitist AS	Dorigo et al.
1995	Ant-Q	Gambardella &Dorigo
1996	Ant Colony System	Dorigo& Gambardella
1996	MAX-MIN AS	Stützle&Hoos
1997	Rank-Based AS	Bullnheimer et al.
1999	ANTS	Maniezzo
2000	Best-Worst AS	Cordón et al.
2001	Hyper-Cube AS	Blum et al.

Three of ACO's main algorithms that are popularly known and successfully implemented are discussed below.

2. Ant System

The first ant algorithm "Ant System" (AS) was developed by Dorigo in the year 1996. He used "Travelling Salesman Problem" (TSP) as its test problem. TSP contains a set of N towns in an area and a salesman. These N towns are interconnected to each other. The problem says that the salesman has to traversed/travel each town of N only once and returning back to the town where it started. This problem is to find the shortest route that a salesman can travel in a short period of time.

Suppose a graph Gr with vertices V and edges E is a connected graph with n numbers of nodes. Let sv be the source and dv be the destination node. Applying Ant System meta-heuristic, the shortest optimum path between sv and dv on the graph Gr can be calculated. The path length in the graph Gr is defined by the number of nodes in the path. Every edges ed(x, y) which belongs to E connects vertices x and y. These edges ed contains a pheromone variable PH(x,y). Every time the ants visit the nodes, the variable PH(x,y) gets a modification in its quantity where every ants deposit their pheromones. The deposition of the pheromone PH(x,y) is giving a clear message indicating that the edge was used by other ants.

The ant which on the node x will use the pheromone variable PH(x,y) of the node y ∈Nx to calculate the probability of finding the next node. Using this probability the ants moves to the next node y. Nx contains all the possible neighbors of x. Px, y is the transition probabilities of node x.

$$\text{Px,y(t)} = \begin{cases} \frac{[PHx,y(t)]^{\alpha}\times[Zx,y]^{\beta}}{\sum_{k\in Nx}[PHx,y(t)]^{\alpha}\times[Zx,y]^{\beta}} & if\ y \in Nx \\ 0, otherwise \end{cases} \quad (1)$$

where, Zx,y = 1/ Dx,y, and Dx,y is the distance between cities x and y. In other words, the shorter the distance between two cities x and y, the higher the heuristic value Dx,y. The role of the parameters α and β is to control the relative importance trail and visibility respectively. If α = 0, the closest cities are more likely to be selected. If on the contrary β = 0, only pheromone amplification is at work. k represents kth ant.

After completing tours by each ant, the pheromone quantity on each edge will be updated according to equation

$$\text{PHx,y(t+1)} = (1-\rho)PHx,y(t) + \textstyle\sum_{k=1}^{m}\Delta PHx,y \quad (2)$$

Where ρ is the coefficient such that (1-ρ) represents the pheromone decay parameter (0< ρ<1) where it represents the trail evaporation when the ant chooses a city and decides to move.

$$\Delta\, PHx,y = \begin{cases} \frac{Q}{Lk}, & if\ (x,y) \in tour\ done\ by\ k \\ 0, & otherwise \end{cases}$$

Where 'm' is the number of ants. L_k is the length of the tour performed by ant k and Q is an arbitrary constant.

AS Algorithm:

Initialize all edges to initial pheromone;
Assign each ant randomly to each city;
for t = 1 to maximum number of city **do**
for k = 1 to maximum number of ants **do**
build a tour by applying the probabilistic transition rule;
end;
if (best tour better than current solution)
update current solution to best tour ;
for every edge **do**
apply pheromone update;
end;
(the next block is optional)
for all edges in best tour**do**
apply additional pheromone increment with Q/length of best tour;
end
end.

3. Ant Colony System

Ant colony system (ACS) is an extension of Ant System. This algorithm was mainly developed in order to improve the efficiency when it is applied to symmetric and asymmetric Travelling Salesman Problems [9][12].

ACS is somewhat different from Ant System in three ways:

- The transition probability rule.
- The local pheromone updating rule.
- The global updating rule which is applied only to the best ant tour edges.

The working of ACS is as follows: m ants are initially positioned on N cities randomly. Each ant travels each city and builds a complete tour by iteratively using the state transition probability rule. During its tour construction, by using the local updating rule, an ant modifies the quantity of pheromone deposited on the edges that is already visited. After every ant completing their tours, the quantity of pheromone deposition on all edges is modified again by using the global updating rule. Using heuristic and pheromone information, all ants are directed to their next step of transition .An edge which contain the highest quantity of pheromone concentration is considered a desirable choice by each ant. The pheromone updating rules both local and global are mainly designed to deposit more pheromone to the edges visited by the ants so to make these edges the most desirable.

ACS State Transition Rule:
In ACS, an ant which is positioned on city x chooses the city u as its next city by using the state transition rule given below;

$$y = \begin{cases} \arg max \left\{ \begin{matrix} [PH(x,u)][Z(x,u)^{\beta}] \\ u \in Jk(x) \end{matrix} \right\} & if\ q \le q_0\ (exploitation) \\ S, & otherwise\ (biased\ exploration) \end{cases}$$
………… (3)

Where q is a random number uniformly distributed in [0 ... 1]. q0 is a parameter ($0 \le q_0 \le 1$), and S is a random variable selected according to the transition probability shown in Eq. (1).The state transition probability rule which results from Eq. (3) and (1) is called pseudo-random proportional rule. This state transition probability rule of ACS gives transitions towards the city connected by short edges and with a large Quantity of pheromone concentration, The parameter q0 determines the relative importance of exploitation versus exploration: Whenever an ant in city x has to choose a city y, a random number ($0 \le q \le 1$) is sampled. So if$q \le q_0$ then the best edge according to Eq. (3) is chosen that is exploitation or else an edge according to Eq. (1) is chosen that is biased exploration.

ACS Global Updating Rule:

Only the globally best ant deposits the additional pheromone in ACS. The ant with the shortest tour from the start of the trial is considered as the global best ant. This choice, together with the use of the pseudo-random-proportional rule, is intended to make the search more directed: Ants search in a neighborhood of the best tour found up to the current iteration of the algorithm. The global pheromone updation is performed after all ants have completed their tours. The pheromone quantity level is updated by using the global updating rule shown below:

$$\mathrm{PH}(\mathrm{x},\mathrm{y}) = (1-\rho).PH(x,y) + \rho.\Delta PH(x,y)$$

where,

$$\Delta PH(x,y) = \begin{cases} (L_{gb})^{-1}, & if (x,y) \in gb^{best} \\ 0, & otherwise \end{cases} \quad (4)$$

$0 < \rho < 1$is the pheromone evaporation parameter, and L_{gb} is defined as length of the globally best (global-best (gb^{best})) tour from the beginning of the trial. It says that only those edges that belong to the globally best tour will be updated with additional pheromone. Instead of gb^{best} another global updating rule which is known as iteration-best can also be used. It also uses the length of the best tour in the current iteration of the trial (L_{ib}). The edges that belong to iteration-best are updated.

ACS Local Updating Rule:

During a construction of tour (solution), all ants visit the edges and change their pheromone quantity level by using the local updating rule shown below:

$$PH(x,y) = (1-\rho).PH(x,y) + \rho.\Delta(x,y) \quad (5)$$

where$0 < \rho < 1$ is a pheromone evaporation parameter.

ACS Algorithm:

Initialize ants
fork=1 to end of iteration **do**
Each ant is assigned on a starting node
fori=1 to maximum number of ants **do**
Each ant usestate transition rule
Local pheromone update
end
Global pheromone update
end

4. MAX-MIN Ant System

Max-Min Ant System (MMAS) is also another extension of Ant system like ACS. It was also developed in order to improve the original Ant System. It was designed and developed by St¨utzle and Hoos in the year 1996. MMAS is also somewhat different from AS[10][11]. Their differences are given below:

- Only the best ant that has the shortest tour length adds pheromone trails or updates the pheromones,
- Explicitly the minimum and maximum values of the pheromone are defined by the designer and are limited. But implicitly limited in AS and ACS.

The probability transition in MMAS is same as that of the original Ant System. The main characteristic of MMAS is that only the best ant can update the pheromone trails. The pheromone update is done using the equation below:

$$\mathrm{PHx,y} = [(1-\rho)\mathrm{PHx,y(t)} + \Delta\mathrm{PHx,y}^{\mathrm{best}}]_{\mathrm{PHmin}}^{\mathrm{PHmax}} \quad (6)$$

Where PH_{max} and PH_{min} are the upper and lower bounds respectively that is set as limitation on the pheromone level. The operator $[z]_b^a$ is defined as

$$[\mathrm{z}]_\mathrm{b}^\mathrm{a} = \begin{cases} \mathrm{a\ if\ z} > a \\ \mathrm{b\ if\ z} < b \\ \mathrm{x\ otherwise} \end{cases} \quad (7)$$

and

$$\Delta\mathrm{PHx,y}^{\mathrm{best}} = \begin{cases} 1/L_{best} & if\ (x,y) \in best\ tour \\ 0 & otherwise \end{cases} \quad (8)$$

Here L_{best} is known as the length of the best ant's tour. It is up to the algorithm designer choice whether it uses the best tour found in the current iteration which is also known as iteration-best (L_{ib}) or the best solution found since the start of the algorithm known as best-so-far(L_{bs}). The designer can also use combination of both. The pheromone updating values are set limited between PH_{min} and PH_{max}.

If $PH_{x,y}$>PH_{max}, $PH_{x,y}$ is set to PH_{max} and if $PH_{x,y}$<PH_{min}then $PH_{x,y}$ is set to PH_{min}. The best ant updates the pheromone using the above equation. In case of Ant System, updation is done to all the edges. Updation is also done in ACS but it is applied only to the edges visited by the best ants.

5. Conclusion

Swarm Intelligence Algorithms are the intelligent techniques that come under the Bio-Inspired Algorithms and are also known as heuristic search method that are capable of solving complex optimization problems. These problems are hard to solve with normal available heuristic techniques. So ACO algorithms are used. But these algorithms have their own weakness.

Applications of these algorithms are not certain and are being explored. Future research should focus on expanding their problem and application areas.

Their structures are somewhat different. The basic weakness is these algorithms are that they do not ensure of giving optimized solutions always. As the problem size increases, the efficiency regarding their execution speed decreases. Various hybrid techniques are also being proposed and developed combining these ant algorithms along with other bio-inspired algorithms (Evolutionary Algorithms). Parallelism is the main factor that leads the efficiency of algorithms high. Parallelism also increases the performance. Intelligent ideas of parallel techniques should be raised.

Defining its starting point and end point earlier is also a weakness of these algorithms. In future the main research area should be on ACO’s theoretical analysis which is still considered difficult. Focusing on application to problems that do not have predefined shape and sizes will also be a good opportunity for future researchers. The convergence time of ACO is uncertain. Typically the solution space of ACO is a construction graph (weighted graph). Future research should also focus on the convergence time and its solution space. Intelligent work is needed for ACO so that it can apply on a set of n-dimensional points.

References

[1] Thakare, S. A. "Comparison of Swarm Intelligence Techniques." International Journal of Computer Science and Business Informatics 1.1 (2013).

[2] Garnier, Simon, Jacques Gautrais, and Guy Theraulaz. "The biological principles of swarm intelligence." Swarm Intelligence 1.1 (2007): 3-31.

[3] Hinchey, Michael G., Roy Sterritt, and Chris Rouff. "Swarms and swarm intelligence." IEEE Computer 40.4 (2007): 111-113.

[4] Abraham, Ajith, He Guo, and Hongbo Liu. "Swarm intelligence: foundations, perspectives and applications." Swarm Intelligent Systems. Springer Berlin Heidelberg, 2006.3-25.

[5] Parpinelli, Rafael S., and Heitor S. Lopes. "New inspirations in swarm intelligence: a survey." International Journal of Bio-Inspired Computation 3.1 (2011): 1-16.

[6] Liu, Yang, and Kevin M. Passino. "Swarm intelligence: Literature overview."Department of Electrical Engineering, the Ohio State University (2000).

[7] Barbagallo, Donato, et al. "A bio-inspired algorithm for energy optimization in a self-organizing data center." Self-Organizing Architectures.Springer Berlin Heidelberg, 2010.127-151.

[8] RakaJovanovic and Milan Tuba, “An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem", Elsvier, Applied Soft Computing, PP 5360- 5366,2011.

[9] Laalaoui Y, Drias H, Bouridah A and Ahmed R B, " Ant Colony system with stagnation avoidance for the scheduling of real time tasks", Computational Intelligence in scheduling, IEEE symposium, 2009, pp 1-6.

[10] David C Mathew, “Improved Lower Limits for Pheromone Trails in ACO", G Rudolf et al(Eds), LNCS 5199, pp 508-517, Springer Verlag, 2008.

[11] Yaseen, SaadGhaleb, and Nada MA AL-Slamy. "Ant colony optimization." IJCSNS 8.6 (2008): 351.

[12] Al Salami, Nada MA. "Ant colony optimization algorithm."UbiCC Journal 4.3 (2009): 823-826.

[13]Gutjahr, Walter J. "A graph-based ant system and its convergence." Future generation computer systems 16.8 (2000): 873-888.

External Links:

- Ant colony optimization algorithms- Wikipedia, the free encyclopedia (http://en.wikipedia.org/wiki/Ant_colony_optimization)
- Swarm intelligence- Wikipedia, the free encyclopedia (http://en.wikipedia.org/wiki/Swarm_intelligence)
- Swarm intelligence- Marco Dorigo and Mauro Birattari (2007), Scholarpedia, 2(9):1462, (http://www.scholarpedia.org/article/Swarm_intelligence)

Author Profile

Keisam Thoiba Meetei received the Bachelor degree in Computer Science and Engineering in 2011 from Shiv Shankar Institute of Engineering and Technology (SSIET), Punjab. He is currently pursuing the Master degree in Computer Science and Engineering from Galgotias University, Greater Noida, UP. His area of Interest is Artificial Intelligence.