ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume - 3 Issue -9 September, 2014 Page No. 8085-8091

Ideas for Some Improvement in A.C.O. On The Behalf Of Presently Working A.C.O.

Er. Anup lal Yadav, Er. Sahil Verma, Er. Kavita
M-Tech Student
Asst. Prof. in C.S.E. Deptt.
EMGOI , Badhauli.
sahilkv4010@yahoo.co.in
Asst. Prof. in C.S.E. Deptt.
EMGOI , Badhauli

Abstract-

Various natural systems teach us that very simple individual organisms can create systems able to solve the complex problems like optimization. Ant Colony Optimization (ACO) is an agent-based technique, which simulates the natural behavior of ants and develops mechanisms of cooperation and learning. Ant Colony Optimization is a metaheuristic approach for solving hard combinatorial optimization problems. The main idea of ACO is to model a problem as the search for a minimum cost path in a graph. Ant Colony Optimization has been successfully applied to scheduling, vehicle routing, and the sequential ordering problems. A review of several Ant Colony Optimization algorithms is done in this paper.

1. INTRODUCTION

Combinatorial optimization problems arise in different fields such as economy, engineering and science. These problems are very hard to solve in practice. Most of these problems are known as NP-hard, which means that there is no algorithm known for solving them in polynomial time. Metaheuristic techniques are now used to solve the hard combinatorial optimization problem [4].

Metaheuristic incorporate concept from different fields such as genetic, biology and Nero-science. Metaheuristic is a set of algorithmic concepts that can be used to define heuristic

methods applicable to a wide set of different problems. Examples of metaheuristic are simulated annealing [1], iterated local search [2] and tabu search [5]. A recent metaheuristic technique is Ant Colony Optimization (ACO), which is inspired by shortest path searching behavior of various ant species. Ant Colony Optimization was introduced by M. Dorigo and colleagues [6]. ACO is used to solve a large number of complex combinatorial optimization problems.

This paper reviews the basis of Ant Colony Optimization algorithms. Section 2 presents the basic concept of real ant colonies which inspired ACO. Section 3 describes several existing ACO Algorithms, while their applications are reviewed in section 4. Section 5 presents the conclusions.

2. REAL ANT COLONIES

Ants are social insects that live in colonies and because of their mutual interaction they are capable of performing difficult tasks. A very interesting aspect of behavior of ant species is their ability to find shortest path between the ants' nest and the food sources [7-9].

Ant are blind, they navigate complex environments and can find food some distance from their nest and return to their nest successfully. They do this by laying pheromones while they navigate their environment. This process, known as stigmergy, modifies their environment to permit communication between the ants and the colony. Ants tend to take the best router between their nest and some external landmark. This natural optimization is again part of stigmergy. As more ants use a particular trail to an external landmark, the trail becomes higher in pheromone concentration. The closet landmark is to nest, the higher the number of round trips made by each ants. For a landmark, that is further away, a lesser number of round trips are made. The higher the concentration of pheromones, the more ants will choose the route over others possible routes. This iterative process achieves suboptimal to optimal trail between the endpoints.

Ant algorithms share some of the fundamental quantities of ant themselves. Ants are cooperative and work collectively towards a common goal. Ant algorithms share these traits in that the simulated ant within environment work in parallel to solve a problem and through stigmergy helps others to further optimize the solution.

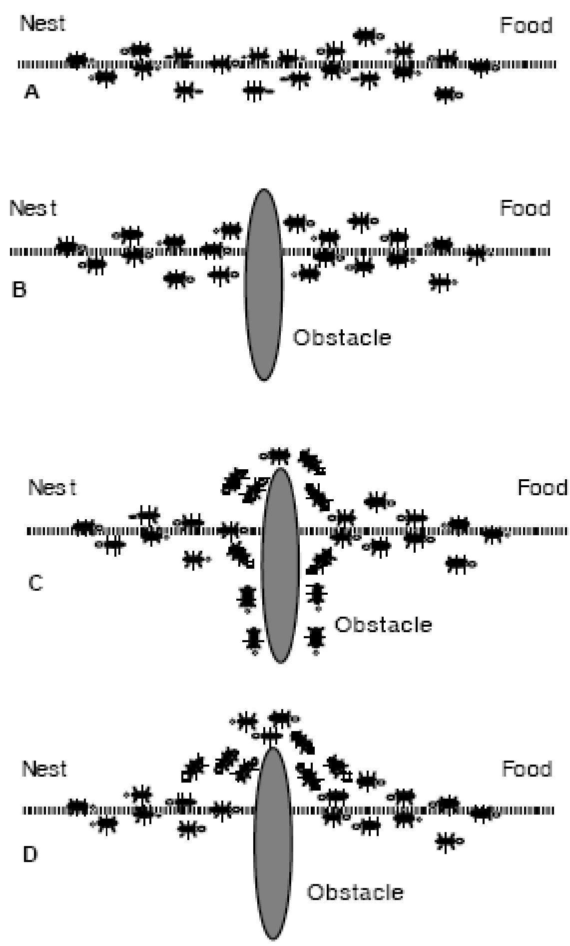

Fig.1. (A) Ants follows a path between nest and food source. (B) An obstacle comes on the path (C) Pheromones deposit more quickly on the shorter path (D) All Ants have chooses shorter path.

Consider figure 1 (A): ants are moving on a straight line that connects a food source to their nest. Ants deposit a certain amount of pheromone while walking and each ant probabilistically prefers to follows a direction rich in pheromone. This elementary behavior of real ants can be used to explain how they can find the shortest path that reconnects a broken line after the sudden appearance of an unexpected obstacle has interrupted the initial path as shown in fig. 1(B). In fact, once the obstacle has appeared, those ants which are just in front of the obstacle can't continue to follows the pheromone trail and therefore they have to choose between the turning left or right. In this situation, we can expect half the ants choose turn right and the other half to turn left. A very similar condition occurred in fig. 1(C). The ants which choose, by chance the shorter path around the obstacle will more rapidly reconstitute the interrupted pheromone trail compared to those which choose the larger path. This, the shortest path will receive a grater amount of pheromone per time unit and in turn a large number of ants will choose the shortest path. Due to this positive feedback process, all ants will choose the shortest path shown in fig. 1(D).

3. ANT COLONY OPTIMIZATION

3.1 General Structure of an ACO Algorithm

The basic operation of an ACO algorithm is as follows: the m artificial ants of the colony move, concurrently. This movement is made according to a transition rule which is based on local information available at the nodes (components).this local information comprises heuristic and memoristic information to guide the search. By moving on the construction graph, ants incrementally build solutions. Ants can release pheromone each time they cross an edge while constructing solutions. Once every ant has generated a solution it is evaluated and it can deposit an amount of pheromone which is a function of the quantity of ants' solution. This information will guide the search of the other ants of the colony in future.

The general operation of ACO algorithm also includes two additional procedures, pheromone trail evaporation and daemon actions. The pheromone evaporation is triggered by the environment and it is used as a mechanism to avoid search stagnation and to allow the ants to explore new space regions. Daemon actions are optional actions- without a natural counterpart to implement tasks from a global perspective that is lacking to the local perspective of ants. Examples are observing the quality of all the solutions generated and releasing an additional pheromone amount only on the transition associated to some of the solutions or applying a local search procedure to the solution generated by the ants before updating the pheromone trails.

General steps for solving a problem by ACO are following:

1. Represent the problem in the form of sets of components and transitions, or by a set of weighted graphs, on which ants can build solutions
2. Define the meaning of the pheromone trails
3. Define the heuristic preference for the ant while constructing a solution
4. If possible implement an efficient local search algorithm for the problem to be solved.
5. Choose a specific ACO algorithm and apply to problem being solved
6. Tune the parameter of the ACO algorithm.

3.2 ACO Variants

Several ACO algorithms have been proposed. Among the available ACO algorithms for NP-hard combinatorial optimization problems are Ant System, Max-Min Ant System, Rank-based Ant System, Best-worst Ant System, Ant-Q and Ant Colony System. We give a short description of theses algorithms. Ant System was the first ACO algorithm. AntNet is a successful ACO algorithm for network routing. This algorithm is rather application specific.

3.2.1 Ant System

Ant System (AS) developed by Dorigo in 1991, was the first ACO algorithm [6, 7]. Initially, three different variants, AS-density, AS-quantity and AS-cycle, differing in the way in which the pheromone trails are updated. In the former two ones, ants releases pheromone while building their solutions, with the difference that the amount deposited in AS-density is constant while the one released in AS-quantity directly depends on the heuristic desirability of the transition. Finally, in AS-cycle, the pheromone deposit is done once the solution is completed.

Ant System is characterized by the fact that the pheromone update is triggered once all ants have completed their solutions and it is done as follows. First, all pheromones trails are reduced by a constant factor. Second, every ant of the colony deposits an amount of pheromone which is a function of the quantity of its solution. Solution in AS are constructed as follows. At each construction step, an ant k in AS chooses to go to a next node with a probability that is computed as

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}}, if \;\; j \in N_i^k, \end{cases}$$

where $\eta_{ij} = 1/d_{ij}$ is a heuristic value that is available a prior, α and β are two parameters which determine the relative influence of the pheromone trail and heuristic information and N_i^k is the feasible neighborhood of ant k when located at node i. By this probabilistic rule, the probability of choosing a particular arc (i, j) increases with the value of the associated pheromone trail τ_{ij} and of the heuristic information value η_{ij}. There is a need to establish a proper balance between the importance of heuristic and pheromone trail information. The pheromone deposit is made once all ants have finished constructing their solutions. First, the pheromone trail associated to every arc is evaporated by reducing all pheromones by constant factor:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij},$$

Where $\rho \in (0,1]$ is evaporation rate. Next, each ant retraces the path it has followed and deposits an amount of pheromone $\Delta\tau_{ij}^{k}$ on each traversed connection:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{k}, \;\; \forall a_{ij} \in S_k$$

Where $\Delta\tau_{ij}^{k}$, the amount of pheromones released is depends on the quality $C(S_k)$ of solution S_k of ant k deposit on the arcs it has visited.

3.2.2 Elitist Ant System

In 1992, a first improvement on Ant System, called Elitist strategy for Ant System (EAS) was developed by Dorigo [6, 7]. In elitist AS, once the ants have released pheromone on the connections associated to their generated solutions, the daemon performs an additional pheromone deposit on the edge belonging to the best solution found. The amount of pheromone deposited, which depends on the quality of that global best solution, is weighted by the number of elitist ants considered, e , as follows:

$$\tau_{ij} \leftarrow \tau_{ij} + e \cdot f\left(C\left(S_{global-best}\right)\right), \ \forall a_{ij} \in S_{global-best}$$

Computational results presented by Dorigo suggest that the use of the elitist strategy with an appropriate value for parameter *e* allows AS finding better solutions and finding them in a lower number of iterations.

3.2.3 Ant Colony System

Ant Colony System (ACS) is one of the first successors of Ant System [7, 13]. It extends Ant System in the following aspects:

1. Ant Colony System uses a different transition rule, which is called pseudo-random proportional rule: let k ants located at node i , $q_0 \in [0,1]$ be a parameter and q is a random value in $[0,1]$. The next node j is randomly chosen according to the probability distribution [7].
2. Only daemon triggers the pheromone update. ACS only considers one single ant, the one generated the global-best solution, $S_{global-best}$.The pheromone update is done by first evaporating the pheromone trails on all the connections used by the global-best ant as follows

   $$\tau_{ij} \leftarrow (1-\rho)\cdot \tau_{ij}, \ \forall a_{ij} \in S_{global-best}$$

   Next, the daemon deposits pheromone by the rule:

   $$\tau_{ij} \leftarrow \tau_{ij} + \rho \cdot f\left(C\left(S_{global-best}\right)\right), \forall a_{ij} \in S_{global-best}$$
3. Ants apply an online step-by-step pheromone trail update that encourages the generation of different solutions to those yet found. Each time an ant travels an edge a_{ij} , it apply the rule:

   $$\tau_{ij} \leftarrow (1-\phi)\cdot \tau_{ij} + \phi \cdot \tau_0$$

   where $\phi \in (0,1]$ is a second pheromone decay parameter. τ_0 is the initial pheromone trail value which is chosen in such a way that , in practice , it corresponds to a lower pheromone trail limit.

3.2.4 Ant-Q

Ant-Q was proposed by Gambardella and Dorigo in 1995 [6, 7]. In practice, the only difference between the Ant Colony System and Ant-Q is the definition of the term τ_0 , which in Ant-Q is set to $\tau_0 = \gamma \cdot \max_{j \in N_i^k} \{\tau_{ij}\}$, where γ is a parameter and the maximum is taken over the set of pheromone trails on the arcs connecting nodes (components) i on which ant k is positioned to all nodes the ant has not visited yet. Setting to a small constant value resulted in a simpler algorithm with approximately the same performance.

3.2.5 Max-Min Ant System

Max-Min Ant System (MMAS) developed by Hoos in 1996 [10, 11]. It extends the basic AS in the following aspects:

- It strongly exploits the best solutions found. The best ant that is allowed to add pheromone may be the iteration-best or the global-best solution. Such a strategy may lead to a stagnation situation in which all the ants follows same solutions, because of the excessive growth of pheromone trails on the areas of a good solution.
- To counteract stagnation effect, a second modification introduced by MMAS is that it limits the possible range of pheromone trail values to interval $[\tau_{\min}, \tau_{\max}]$.
- The pheromone trail are initialized to the upper pheromone trail limit, with a small pheromone evaporation rate, increase the exploration of solutions at the start of the search.
- In MMAS, pheromone trails are reinitialized each time the system approaches stagnation or when no improvement tour has been generated for a certain no. of consecutive iterations.

3.2.6 Rank-based Ant System

The rank-based Ant System (AS_{rank}) [14] is another extension of the Ant System proposed by Bullnheimer et.al. in 1997. In AS_{rank} , each ant deposit an amount of pheromone that decrease with its rank. Additionally, as in EAS, the best-so-far ant always deposits the largest amount of pheromone in each iteration. The idea of ranking into pheromone update is as follows:

1. The m ants are ranked according to decreasing quality of their solutions: $(S'_1, S'_2, ..., S'_m)$, with S'_1 being the best solution built in the current generation.
2. The daemon deposit pheromone on connections passed by $\sigma - 1$ the best ants. The amount of pheromone deposited directly depends on the ants' rank and on the quality of its solution.
3. The connections crossed by global-best solution receive an additional amount of pheromone which depends on quality of that solution. It receives a weight of σ .

This operation made is put into effect by means of the following pheromones update rule, which is applied to every edge once all the pheromone trails have been evaporated:

$$\tau_{ij} \leftarrow \tau_{ij} + \sigma \cdot \Delta\tau_{ij}^{gb} + \Delta\tau_{ij}^{rank}$$

3.2.7. Best-Worst Ant System

Best-Worst Ant System (BWAS) proposed by Cordon et. al. in 1999 is an ACO algorithm which incorporates evolutionary computation concepts [13, 15]. It constitutes another extension of Ant System, which uses its transition rule and pheromone evaporation mechanism. Besides, as done in MMAS, Best-Worst Ant System always considers the systematic exploitation of local optimizers to improve the ants' solutions. In BWAS, the three following actions are found:

1. The best-worst pheromone trail update rule, which reinforces the edge contained in the global best solution. The update rule penalizes every connection of the worst solution generated in the current iteration. Hence, BWAS update rule becomes:

$$\tau_{ij} \leftarrow \tau_{ij} + \rho \cdot f\left(C\left(S_{global-best}\right)\right),$$

$$\forall a_{ij} \in S_{global-best}$$

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij}, \forall a_{ij} \in S_{current-worst} \ \& \ a_{ij} \notin S_{global-best}$$

2. A pheromone trail mutation is performed to introduce diversity in the search process. To do so, the pheromone trail associated to one of the transitions starting from each node is mutated with probability by considering any real coded mutation operator.
3. BWAS considers the re-initialization of the pheromone trails when it gets stuck, which is done by setting every pheromone trail to τ_0 .

3.2.8. AntNet

AntNet is an extension of single Ant Colony System algorithm. AntNet is closer to the real ants' behavior that is inspired the development of the Ant Colony Optimization algorithm for NP-hard problems [7]. The main features of the algorithm are as follows:

1. At regular intervals, artificial ants are moving towards destination nodes selected according to the traffic distribution.
2. Artificial ants act concurrently and independently and communicate in an indirect way, through pheromone they read and write locally on the nodes.
3. Each artificial ant searches for a minimum cost path joins its source and destination node.
4. Each artificial ant moves step-by-step towards it destination node. At each intermediate node a greedy stochastic policy is applied to choose next node.
5. While moving, the artificial ants collect information about to time length, the congestion status and the node identifier of the followed path.
6. Once they have arrived at the destination, the artificial ants go back to their source node by moving alone the same path as before but in opposite direction.
7. During this backward travel, node-local models of the network status and the pheromone stored on each visited node are modified by the artificial ants as a function of the path they followed.
8. Once they have returned to their source node, the artificial ants are deleted from the system.

4. APPLICATIONS

Ant Colony Optimization algorithms have been applied to a large number of different combinatorial optimization problems. Current Ant Colony Optimization applications fall into two classes of application. The first class of problems comprises NP-hard combinatorial optimization problem, for which classical techniques often show poor behavior. ACO applications to these problems are that ants are coupled with local search algorithms. The second class of application comprises dynamic shortest path problems, where the problem instance under solution changes at algorithm run time. These changes may affect the topology of the problem such as the availability of links very with time. This class comprises applications of ACO to routing in telecommunication. networks.

The first combinatorial problem tackled by an ACO algorithm was Travelling Salesman Problem (TSP), because this problem is probably the best known instance of an NP-hard, constrained shortest path problem [9]. The next two applications were the Quadratic Assignment Problem (QAP) [16] and Job-Shop Scheduling Problem [17]. The next applications are the first network routing applications. ACO applications include classical vehicle routing problems [18], sequential ordering [19], and graph coloring problems [20]. ACO metaheuristic is used to solve a large number of combinatorial optimization problems such as generalized assignment, multiple knapsack and constraint satisfaction problems. ACO recently was used for machine learning, fuzzy logic rules [21] and Bayesian networks [22].

Now, ACO has been applied to several problems such as Quadratic Assignment Problem, Sequential ordering, vehicle routing, scheduling and packet-switched network routing. ACO is applied to novel real world problems with good results.

5. CONCLUSIONS

Ant Colony Optimization is a well defined and good performing metaheuristic technique that is applied to solve the complex combinatorial problems. Ant Colony Optimization is a population-based metaheuristic which exploits an indirect form of memory of previous performance. In this paper, we have reviewed the ideas of this approach that lead from the biological inspiration to ACO metaheuristic. Most of the existing approaches have been described. The main difference between the various Ant System extensions consist of the techniques used to control the search process.

REFERENCES

[1] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi, “Optimization by Simulated Annealing,” Science-New Series, Vol. 220, pp. 671-680, 1983

[2] E. H. L.Aarts and J. K. Lenstra, “Local Search in Combinatorial optimization,” John Wiley & Sons, Chichester, 1997.

[3] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning,” Addison – Wesley, 1989.

[4] S. Voss, S. Martello and I.H. Osman, “Meta-Heuristics: Advances and trends in Local Search Paradigms for Optimization,” Kluwer Academic Publishers, Boston, MA, 1999.

[5] F. Glover and M. Laguna,,“Tabu Search,” Kluwer Academic Publishers, Boston, MA, 1997.

[6] M. Dorigo, V. Maniezzo, and A. Colorni, “The Ant System: Optimization by a colony of cooperating agents,” IEEE transcations on Systems, Man, and Cybernetics, vol. 26, pp. 29-41, 1996.

[7] M.Dorigo and T. Stutzle, “Ant Colony Optimization,” MIT Press, Cambridge, MA, 2004.

[8] E. Bonabeau, M. Dorigo, and G. Theraulaz, “Swarm Intelligence: From Natural to Artificial Systems,” Oxford University Press, New York, 1999.

[9] M. Dorigo and T. Stutzle,, “ACO algorithms for the travelling salesman problem,” John Wiley & Sons, Chichester, UK, 1999.

[10] T. Stutzle and H. H. Hoos, “MAX-MIN Ant System,” Future Generation Computer Systems, vol. 16, pp.889-914, 2000.

[11] T. Stutzle and H. H. Hoos, “The MAX-MIN Ant System and local search for the travelling salesman problem,” IEEE International Conference on Evolutionary Computation, pp. 309-314, Piscataway, 1997

[12] T. Stutzle and H. H. Hoos, “Improving the Ant System: A detailed report on the MAX-MIN Ant System,” Technical Report AIDA-96-12, Germany, 1996

[13] Oscar Cordon and T. Stutzle, “A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends,” Mathware and Soft Computing, vol. 9, 2002

[14] B. Bullnheimer and C. Strauss, “A New rank-based version of the Ant System: A computational study,” Central Joournal for Operations Research and Economics, vol. 7, pp.25-38, 1999.

[15] Oscar Cordon, I. Frandez and L. Moreno, “A new ACO model integrating evolutionary computation concepts: the best-worst Ant System,” Proceedings of ANTS2000, pp.22-29, Belgium, 2000

[16] M. Dorigo and A. Colorni, “the Ant System applied to the quadratic assignment problem,” Technical Report IRIDIA, Belgium, 1994.

[17] T. Stutzle, “An Ant approach to the flow shop problem,” European conference on Intelligent Techniques and Soft Computing, vol. 3, pp. 1560-1564, Germany, 1998

[18] L.M. Gambardella and G. Agazzi, “A multiple ant colony system for vechicle routing problems with time windows,” New Ideas in Optimization, pp. 63-76, McGraw Hill, London, UK, 1994

[19] M. Dorigo and L.M. Gambardella, “Ant Colony System hybridized with a new local serach for the sequential ordering problem, Journal on Computing, vol. 12, pp.237-255, 2000.

[20] D. Costa and A. Hertz, “Ants can color graphs,” Journal of the Operational Research Society, vol. 48. pp.295-305, 2000.