PUNE RESEARCH DISCOVERY ISSN 2455-9202
AN INTERNATIONAL JOURNAL OF ADVANCED STUDIES VOL 6, ISSUE 2

BIO-INSPIRED COMPUTING AS OPTIMIZATION TECHNIQUE USING ANT COLONY OPTIMIZATION

PRIYA GODBOLE
Assistant Professor,
Prerna College of Commerce, Nagpur,
(MS) INDIA

SNEHAL DONGARE
Assistant Professor,
Prerna College of Commerce, Nagpur,
(MS) INDIA

ABSTRACT

Nature is the most intelligent entity and its techniques and methods for solving most complex problems are extremely massive, amazing and beyond our imagination. Researchers have inspired to imitate nature to solve hard and complex problems in computer sciences as well. Many optimization techniques and problem-solving methodologies have been taken inspiration from natural processes and come up with great natural solution. From ancient times, nature itself considered as a supreme source of motivation for mankind; so it was very common for mankind to turn to nature for answers when faced with challenges. This paper proposes how ant colony optimization (ACO) algorithm can be used for solving optimization problems and how it can be improved to solve routing problems.

Keywords: *Ant Colony Optimization(ACO), pheromone, foraging, Travelling salesman problem(TSP), Dijkshtra algorithm, network routing, ACO routing algorithm*

INTRODUCTION

Ant Colony Optimization is an optimization technique from the class of Bio-Inspired Computing that is commonly known as *Swarm Intelligence*. It is artificial intelligence technique based on collective and self-organised behaviour of insects and other animals. Ant colony optimization is a class of optimization algorithms modelled on the actions of an ant colony.

The path on which ants started walking from food source and return back to source; deposit certain substance on the ground called *pheromone*. On perceiving, presence of pheromone;

other ants tend to follow paths where pheromone concentration is higher. By using this mechanism, ants are able to transport food to their nest in an effective way.

This biologically inspired approach can be used as optimization technique and implemented for solving routing problems [1] and wireless sensor networks [2] in networking and many other areas also. Due to the characteristics of self-organization, auto-optimizing, an ant colony algorithm is used in Ad Hoc network, wireless sensor network, and so on. Ant colony algorithm used in routing is improved continuously.

This paper gives an overview of improved ant colony optimization algorithm to find the optimal path by routing taking inspiration from Dijkshtra algorithm.

1. The Optimization Technique

In Ant Colony Optimization(ACO), a number of artificial ants build solutions to the any optimization problem and exchange information on the basis of goodness of these solutions via a communication scheme that reminds the one adopted by real ants. Different ant colony optimization algorithms [9] have been proposed to find solution to discrete optimization problems related to computer science.

a) **Applying ACO to continuous optimization:** Practically many optimization problems can be formulated as continuous optimization problems. These problems are described by the fact that the decision variables have continuous domains, in comparison to the discrete domains of the variables in non-continuous optimization. While ACO algorithms [4] were developed initially to solve discrete problems, but now a day's these algorithms are adapted to solve continuous optimization problems and overjoyed the attention of many researchers.

b) **Algorithms inspired by Foraging and Path Marking:** In these algorithms, the search space in the graph is explored by the ants by depositing the pheromone on the edges they already visited, to coordinate to other ants. Contrary to ACO [1] , in these algorithms the pheromone directs the ants toward unexplored areas of the search space. The aim is to cover whole graph by visiting each node, without knowing the graph physical arrangement. Another example of algorithm inspired by ants' path marking is a search algorithm [4] for continuous optimization problems that was inspired by the foraging behaviour of ants.

c) **ACO for the Travelling Salesman Problem:** Ant colony optimization is an iterative algorithm [3]. In the travelling salesman problem [8], a set of cities is given and the distance between each of them is known. It aims to find and arrange the shortest tour that allows the salesman to visit each city only once. In ant colony optimization, the problem is take up by simulating a number of artificial ants moving on a graph that encodes the problem itself: each vertex represents a city and each edge represents a connection path between two cities. A variable called pheromone is associated with each edge and can be perceived and modified by ants. Number of artificial ants tends to build a solution by walking from one vertex to other on the graph with the restriction of not visiting any previously visited vertex. At each step of the solution construction, an ant selects the next vertex that is to be visited according to a stochastic mechanism which biased by the variable pheromone: If current position is in vertex *i*, the next vertex is selected stochastically among the previously unvisited ones. At any instance, if *j* has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge *(i, j)*. This is shown in Figure 1.

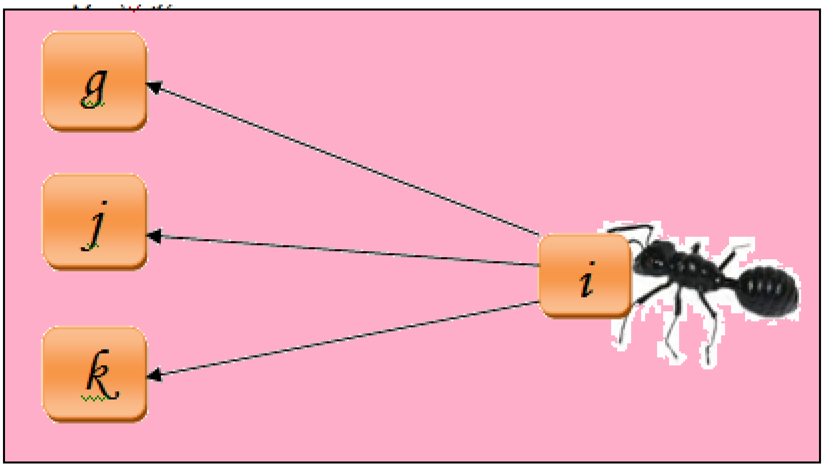

Fig:1 An ant in city i chooses the next city to visit by stochastic mechanism: if j has not been previously visited, it can be selected in proportion to high pheromone concentration [3]

On completion of first iteration, the quality of the solutions constructed by the ants is measured; the pheromone values are modified in order to bias ants in upcoming iterations to construct solutions similar to the best ones constructed before.

d) **Dynamic Optimization Problems:** Dynamic problems are featured by the fact that the search space changes with time. While searching, the conditions of the search may change, and hence the definition of the problem instance and the quality of the solutions already found may change. In such a situation, it is extremely important that the algorithm should be able to adjust the search direction, following the changes of the problem being solved. An example of dynamic optimization is routing in telecommunication networks [3]. ACO algorithms have also been applied to dynamic versions of the Travelling Salesman Problem, where either the distance between pairs of cities changes, or new cities are dynamically added or previous cities are removed from the set of cities to be visited.

e) **ACO in Telecommunication Networks:** Ant Colony Optimization 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, vary over time. ACO algorithms [9] were applied initially to routing problems in telephone switching networks and then in packet-switched networks.

f) **Stochastic Optimization Problems:** Apart from the network routing problems, for which the main focus was put on their dynamic character, the stochastic travelling salesman problem (TSP) [8] was the first stochastic problem tackled by ACO algorithms. In the TSP, each city has assigned a probability of requirement to visit and aims to find a tour of shortest possible length over all the cities, with taking into account the strategy of visiting a random subset of cities in the order same as they appear in the a priori tour.

2. Model Assumptions:

In Ant Colony Optimization (ACO), [1] an artificial ant tends to builds a solution by traversing the fully connected construction graph G (V, E), where V is a set of vertices and E is a set of edges. G can be obtained from the set of solution components C in two ways: components can be represented either by vertices or by edges. Artificial ants move from one vertex to other along the edges of the graph, results in building a partial solution. In addition to this, ants deposit a certain amount of pheromone on the solution components; that is, either on the vertices or on the edges that they traverse. The quality of the solution found may depend on the amount (**τ**) of pheromone deposited. Foregoing ants use previously deposited pheromone information as a guide to explore promising regions of the search space.

The construction graph [3] can be obtained by representing solution components as vertices on which pheromone is deposited.

The heuristic information η_{ij} [3] is only related to the distance to the next node, which can be calculated by:

$$\eta_{ij} = \frac{1}{d_{ij}}$$

where d_{ij} denotes the distance between node i and the next node j.

The ACO metaheuristic is shown in Algorithm 1. After initialization, the metaheuristic iterates over three phases: at each iteration, a number of solutions are constructed by the ants;

these solutions are then improved through a local search (this step is optional), and finally the pheromone is updated.

Algorithm 1 The Ant Colony Optimization
Set parameters, initialize pheromone trails
while termination condition not met **do**
ConstructAntSolutions
ApplyLocalSearch (optional)
UpdatePheromones

Endwhile

I. **Pheromone Initial Optimization**. Node number m and communication radius R are determined. Let $V_i = \Phi$, $i = 1, 2, \ldots, m$, be the set recording the previous node number and $P_i = 0$ the node coefficient. Let d_{ij} be the traffic demand between the node i and the node j.

Step 1. End point communicates with adjacent nodes. The adjacent nodes P ($k = 1, 2..., m$) record the number of termination point and update node coefficient. If d_{o2k} is less than the R, the P_k is set as one and O_2 is put into the set V_k.

Step 2. Get the number S of nodes which could directly communicate with starting point. The initial value of S is zero, and the value plus one when d_{o1k} is less than the R.

Step 3. Node j communicates with adjacent node k. When $P_j < P_k$ or $P_k = 0$, the node k updates p and records number of node j until start point gets number of its adjacent nodes ($j = 1, 2..., m$). Through this operation, the node gets the information about routes from the initial node to the end point.

II. **Construction of Solution:** Simulated with the search food process of ant colony, the model for basic ant colony algorithm is as follows.

Suppose S is a set and $s_1, s_2, \ldots$, are subsets and coverage of S and, at the initial moment, they are selected; ant will be randomly placed on the m-subsets, assuming that the initial information of each subset $\tau_{ij}(0) = C$. The probability to transfer of ant k from the subset i to subset j is:

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{t\in \text{allowed}_t}[\tau_{it}]^{\alpha}[\eta_{it}]^{\beta}}, & j \in \text{allowed}_t \\ 0, & \text{otherwise.} \end{cases}$$

Among them, t is the iteration number, k is ID ($k = 1, 2, \ldots, m$) for ants, and t is the iteration number; allowed t is the next subset selected from ant k; $\tau_{ij}(t)$ is the pheromone strength from subset i to subset j; η_{ij} is the inspired degree of ant k shifted from subset i to the subset j. These two parameters α and β are accumulation of information and inspired information in the process of ant's sports, reflecting the relative importance of ants to choose the next subset.

In order to maintain the equilibrium of energy consumption of nodes, improved ant colony algorithm uses the energy factor based on the basic ant colony routing algorithm [2] to find shorter and high energy path. Then, the improved probability to transfer of ant k from node i to node j is defined as follows:

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}[\xi_{ij}]^{\chi}}{\sum_{t\in \text{allowed}_t}[\tau_{it}]^{\alpha}[\eta_{it}]^{\beta}[\xi_{it}]^{\chi}}, & j \in \text{allowed}_t \\ 0, & \text{otherwise.} \end{cases}$$

Using the relative factor represents normalization of method, where ξ is relative energy factor which equals residual energy of node j divided by the initial node energy. χ are the parameter that represents accumulation of information and inspired information in the process of ant's traversing, it reflects the relative energy consumption by ants that help to choose the next subset.

III. **Pheromone Update**. Subset p is determined, and ant will stop when elements that selected subset contained target node; this will mark the end of the cycle. After all the ants have completed first iteration, the pheromone is adjusted according to the following equations:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \sum_{k=1}^{m}\Delta\tau_{ij}^{k},$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k}, & \text{if } j \in \text{solution of ant } k \\ 0, & \text{otherwise.} \end{cases}$$

Among them, $(1-\rho)$ is the attenuation coefficient of the pheromone; usually $\rho \in [0, 1]$ to avoid unlimited accumulation of informational on subset; L_k is the number of subsets and k selected in this cycle; Q is the pheromone strength; it affected the convergence speed of algorithm to a certain extent.

3. Ant Colony Routing Algorithm

a. Dijkshtra Algorithm : Dijkstra algorithm was proposed by computer scientist Edsger Wybe Dijkstra to find the shortest path from an original point to other points in the directed graph. Every point updates the shortest route information from the initial point. When a route between two nodes is found, the weight of that route is updated in the table. When a shorter path is found, its new optimal weight will be updated to the table, by replacing the old value. The algorithm [2] allows traffic to be routed around the network at the same time connecting to the least number of nodes as possible. The system works but does not take into account the flood of traffic and load balancing.

b. Ant Colony Optimization Algorithm : Ant Colony Optimization algorithms uses virtual pheromone tables much like when an ant follows a path dropping pheromones to reinforce it. As rapidly the ants move down a path, the better the throughput of ants, and thus the higher concentration of pheromone. Similarly, the pheromone tables in ACO algorithms [9] allow fast routes to score a maximum chance of being selected although the less optimal route scores a low chance of being selected.

In most ACO routing algorithms [2], the quality of a route is measured from its number of hops and communication delay. Hence, if number of hops are smaller on a path and communication delay is shorter then; it gets more pheromone deposited which helps to transfer data packets with a higher probability. Although, the difficulty with ACO routing algorithms in communication networks is that the amount of pheromone is updated some time before it is utilized in sending packets and link disconnections frequently occur because of movement of nodes. However, even though a path with a small number of hops and short communication delay has high pheromone deposited, the path may become unavailable quickly when link is disconnected.

To solve this problem, we propose an improved ant colony optimization (ACO) technique with a new pheromone updating rule.

c. Improved Ant Colony Routing Algorithm: In basic ant colony algorithm, wireless sensor network could be described as an undirected graph. At the beginning, the insufficient amount of initial pheromone generates low solving speed and high consumption, which has affected the overall performance of ant colony algorithm. In order to solve the problem, the improved ant colony algorithm based on Genetic-Ant Colony algorithm is used. However, it might cause the breaks in data transmission after crossover and mutation. In the paper, the idea has been proposed about Dijkstra algorithm to improved ant colony algorithm [2] distributing the nodes into the directed graph. It could increase the original efficiency of an algorithm and ensures its stability.

In routing problem, a solution can be represented through a set of n variables, where n is the total number of nodes. A pheromone value is related with each possible solution component. The variable X_i indicates next hop node in network after node i. Here solution component is a path from current node to next hop node. The pheromone value τ_{ij} is associated with the solution component c_{ij}, which consists of the assignment v_i^j.

Here solution component is a pair of nodes that come one after other in search space while routing. It is observed that the construction graph could be obtained by representing solution components as vertices to which pheromone is deposited. Figure 3 shows the construction graph for routing where 4 nodes are available. In this figure component solutions are represented with (a) edges and with (b) vertices.

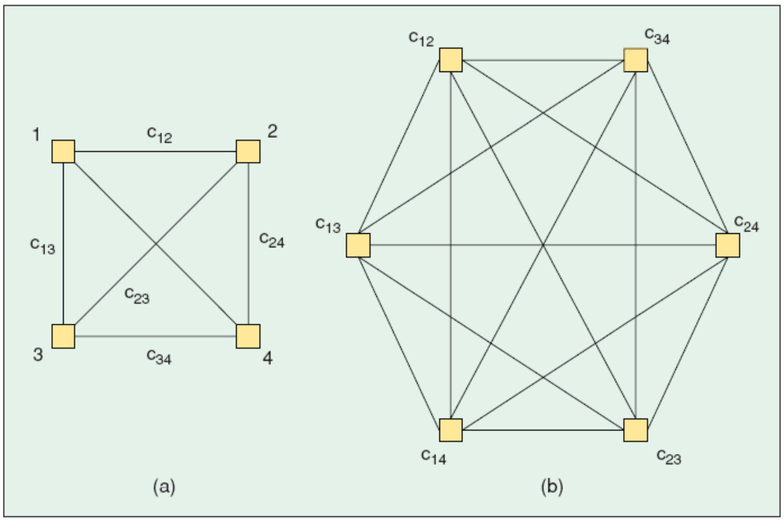

Fig:3 Possible Construction Graph for routing with 4 nodes where components represented with (a) the edges and (b) the vertices [3]

Simulation Result:

After comparison with basic ACO algorithm, the improved ant colony routing algorithm is more efficient. The popular Dijkstra algorithm of routing was selected as reference to improve basic ACO algorithm to reduce average routing path in networks and helps to find shortest path.

CONCLUSION:

The algorithms based on foraging behaviour of ants provide natural solutions to discrete optimization problems and telecommunication networks. Compared with basic ant colony routing algorithm, improved ant colony routing algorithm is an algorithm with low energy consumption and has high performance. An improved ant colony routing algorithm, which is inspired by the Dijkstra algorithm changing the wireless sensor network undirected graph to directed graph to improve the ant colony algorithm. The improved ant colony algorithm also reduces the energy consumption. We can see that the improved ant colony routing algorithm has more optimum routing path that guarantees the message proper flow.

REFERENCES

[1] A Review on Ant colony optimization: Introduction and recent trends, Christian Blum, *ALBCOM, LSI, Universitat Politècnica de Catalunya, Jordi Girona 1-3, Campus Nord, 08034 Barcelona, Spain*Accepted 11 October 2005 Communicated by L. Perlovsky

[2] A Clustering Routing Algorithm Based on Improved Ant Colony Optimization Algorithms for Underwater Wireless Sensor Networks By Xingxing Xiao and Haining Huang

https://doi.org/10.3390/a13100250

[3] Ant Colony Optimization, IEEE Computational Intelligence Magazine · December 2006

[4] A. Bauer, B. Bullnheimer, R. F. Hartl and C. Strauss, "An ant colony optimization approach for the single machine total tardiness problem", *Proc. 1999 Congr. Evolutionary Computation*, pp. 1445-1450, 1999.

[5] S. E. Elmaghraby, Activity Networks, New York:Wiley, 1977.

[6] Dorigo M, Blum C. Ant colony optimization theory: A survey. Theoret Comput Sci 2005;344(2–3):243–78

[7] Dorigo M, Stützle T. Ant Colony optimization. Cambridge, MA: MIT Press; 2004

[8] Guntsch M, Middendorf M. Pheromone modification strategies for ant algorithms applied to dynamic TSP. In: Boers EJW, Gottlieb J, Lanzi PL, Smith RE, Cagnoni S, Hart E, Raidl

GR, Tijink H, editors. Applications of evolutionary computing: Proceedings of EvoWorkshops, 2001. Lecture Notes in Comput Sci, vol. 2037. Berlin: Springer; 2001. p. 213–22

[9] Korošec P, Šilc J, Robic B. Mesh-partitioning with the multiple ant-colony algorithm. In: Dorigo M, Birattari M, Blum C, Gambardella LM, ˇ Mondada F, Stützle T, editors. Proceedings of ANTS 2004—Fourth international workshop on Ant colony optimization and swarm intelligence. Lecture Notes in Comput Sci, vol. 3172. Berlin: Springer; 2004. p. 430–1

[10] Merkle D, Middendorf M. Modelling the dynamics of ant colony optimization algorithms. Evolutionary Comput 2002;10(3):235–62.