Optimal-Ant Colony Optimization in Swarm Intelligence

Anita
Department of Computer & Science & Engineering
MRIU, Faridabad
anidagar@gmail.com

SS Tyagi
Professor and Head
Department of Computer Science & Engineering
MRIU, Faridabad
shyamtyagi@hotmal.com

Abstract: *To describe the approach of real-world activities we have proposed an algorithm and its diagram for the Optimal Ant Colony Optimization Technique. In this paper we are describing the techniques of swarm intelligence. In Ant Colony Optimization Approach we will check through the algorithm whether the appropriate path is selected to reach the destination. To explain the concept of this algorithm we have used a real-world example of Ants. To optimize the path in the search space, we have proposed an OACO algorithm.*

1. INTRODUCTION

Swarm Intelligence is defined as a technique that focuses on the collective behaviors of the social insects that result from the local interactions of the individuals with each other and with their environment[1]. SI is an innovative computational and behavioral metaphor for solving distributed problems that takes its inspiration from the behavior of social insects and the swarming, flocking, herding, and shoaling phenomena of vertebrates[9].

Examples: colonies of ants, flocks of birds, herds of animals and schools of fish. Among many successful bio-inspired swarm intelligence computational paradigms, two well-known approaches are Ant Colony Optimization (ACO) [7] and Particle Swarm Optimization (PSO) [4]. OACO (Optimal Ant Colony Optimization) is the algorithm that we are proposing for optimization. Ant Colony Optimization (ACO) is one of the techniques of swarm intelligence.

ACO is used to find the path through the graphs in which neurons will act as the vertices and there are connected through the directed edges[9]. The ACO algorithm can be applied in solving number of different problems; the best example in which we use the ACO is Travelling Salesman Problem.

Other similar application examples include the Sequential Ordering Problem and the Vehicle Routing Problem. ACO is also used in solving the Quadratic Assignment Problem, as well as telecommunication network problems including routing.

Based on these ideas, in this paper we have proposed the Optimal Ant Colony Optimization (OACO) algorithm to train the network. We will find the shortest path through the graph with the help of this algorithm. The paper is organized as follows. Section II gives the introduction about Ant Colony Optimization. Related work is introduced in section III. Section IV discusses the OACO method. Finally conclusion is given in the section V.

2. ANT COLONY OPTIMIZATION

ACO is a technique for solving combinatorial optimization problems that can be viewed in the form of path finding problem. ACO technique is inspired by the way how real ants find shortest paths from their nest to food i.e. from source to the food destination[1]. An essential part from which we get the idea is the indirect communication of the ants with the help of pheromone.

Pheromone is a chemical substance that is released by an ant on the pathway while moving from nest to the food source and that shows the behavior of other individuals of the same species[1]. So, Ants lay the pheromone on the way to mark the paths to their food sources. The pheromone traces are then smelled by other ants and which lead them to the food source[1].

A biological experiment called the double bridge experiment was the inspiring source for the first ACO algorithm[7]. In the experiment a double bridge with two branches of different lengths are connected to the nest of the ant and to the food source[7]. The Long Branch is twice in length than the shorter branch. When the experiment was run, after watching for few minutes it was noticed that after a few minutes almost all ants move to the shorter branch[7]. This behavior was interesting because this experiment was carried out on the Argentine ants and these ants cannot see very well. The behavior of ants shows that the ants lay pheromone along their path.

3. RELATED WORK

Global search techniques, with the ability to broaden the search space in the attempt to avoid local minima, has been used for connection weights adjustment or architecture optimization of MLPs, such as ant colony optimization (ACO) [10], particle swarm optimization (PSO) [5]. A genetic algorithm was hybridized with local search gradient methods for the process of MLP training (weight adjustment) in [3].

In [10], ant colony optimization was used to train the weight of a fixed-topology MLP. Pillai, K. G. explains a novel overlapping swarm intelligence algorithm is introduced to train the weights of an artificial neural network. Training a neural network is a difficult task that requires an effective search methodology to compute the weights along the edges of a network[11]. Marco Dorigo and Mauro Birattari defines Swarm intelligence as the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization[7].

4. THE OACO METHOD

The OACO method is known as Optimal- Ant Colony Optimization method. In this method we have shown that how ants will move through the graph to search the food.

A. OACO algorithm

In OACO method we will start the process by initializing the nest equals to n, number of ants will be i, number of paths will be j and initially we will fix the pheromone amount equals to zero.

ACO is used to allocate training iterations among a set of candidate network topologies. The desirability in ACO is defined as:

$$d(i) = \frac{1}{h+1}$$

Where h is the number of hidden nodes in neural network[1].

We will select the edge based on the amount or level of the pheromone deposit by the ants on the path they traverse. Ant k will move from state I to j with the probability of $p^k(ij)$. This probability is based on the amount of pheromone deposit on the path.

The OACO algorithm

```
1.  for (Ant i=1; i<=n; i++)
2.  float pheromone = 0.5;
3.  int nest=n;
4.  for (pheromone j=; j<m; j++)
5.      { // initially no pheromone
6.      j = j++   //on each path
7.      }
8.  end for;
9.  Memory(A,j)
10. {
11.     if(j[ i1] > j [ i2])
12.     then
13.     Choose path to the food
14.     else
15.     Choose the other path
16. }
17. If (destination==food source)
18.     then
19.     Return and update the pheromone
20. else
21.     Keep on searching
22. end for
```

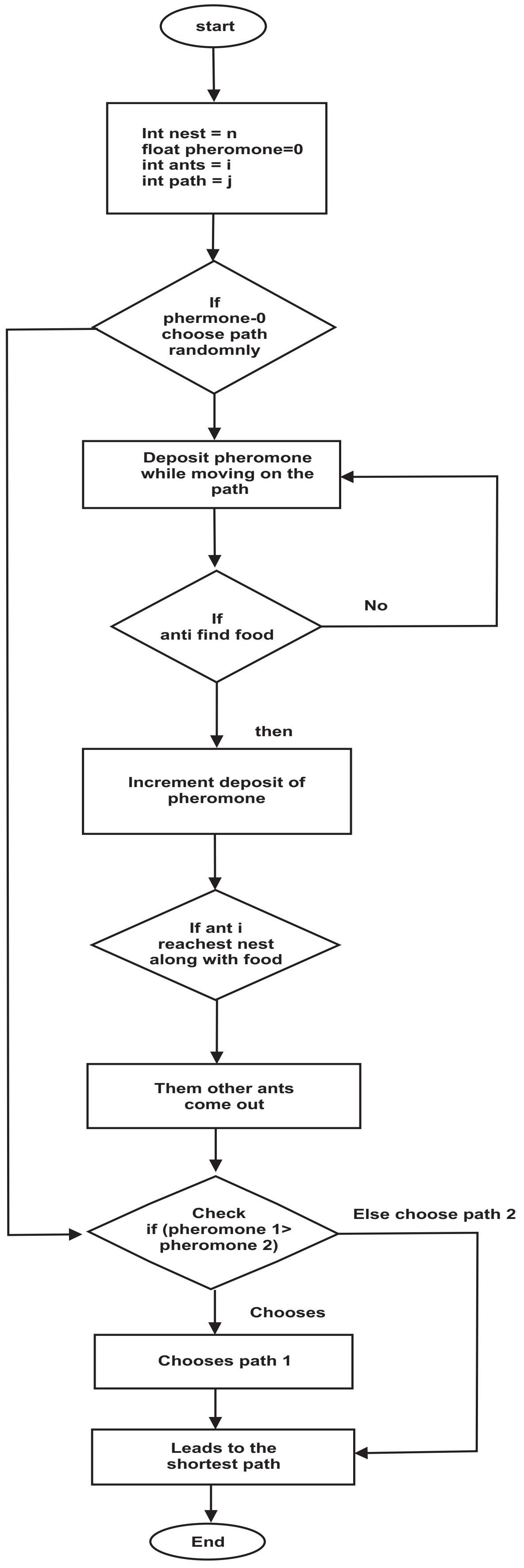
start
Int nest = n
float pheromone=0
int ants = i
int path = j
If phermone-0 choose path randomnly
Deposit pheromone while moving on the path
If anti find food
No
then
Increment deposit of pheromone
If ant i reachest nest along with food
Them other ants come out
Check if (pheromone 1> pheromone 2)
Else choose path 2
Chooses
Chooses path 1
Leads to the shortest path
End

Fig. 1. Working of the OACO mMethod.

We have in figure 1.1,

nest== n

float pheromone=0 // initially

number of ants will be i and number of paths will be j. Then first ant will come out of the nest , since the pheromone=0 initially, the ant will randomnly choose any path. Ants will deposit the pheromone while moving on the path. If ant i finds the food then they will return taking food and increment the level of deposit of pheromone on the path. If unable to find the food, they will keep on searching the food. If ant I reaches the nest after taking the food then others ants will come out from the nest. Now, ants will compare the pheromone deposits on the path. If (pheromone[j1] > pheromone[j2]) ,they will choose the path 1 else path 2. Similarly they will follow the concept, and slowly they will lead to the shortest path.

5. CONCLUSION

In this paper, a OACO algorithm is proposed to generate the shortest path through the graph. We have taken the example of ants to explain the algorithm. By utilizing the OACO algorithm and simple ACO concept, this algorithm provides an efficient method for generating the shortest path. The OACO method can be applied to lots of real world tasks.

REFERENCES

1. Anita and Dr. S. S. Tyagi, "Provoding the security for the building using Ant Colony Optimization Technique", International journal of Scientific and Research publications, Volume 3, issue 6 , June 2013.

2. C. Blum and K. Socha, "Training feed-forward neural networks with ant colony optimization: An application to pattern classification", Fifth International Conference on Hybrid Intelligent Systems (HIS'05), pp. 233-238, 2005.

3. E. Alba and J.F. Chicano, "Training Neural Networks with GA Hybrid Algorithms", K. Deb(ed.), Proceedings of GECCO'04, Seattle, Washington, LNCS 3102, pp. 852-863, 2004.

4. Haza Nuzley A. Hamed, Siti Mariyam and Naoimi salim: "Particle Swarm Optimization for Neural Network learning enhancement"; Jurnal Teknologi, 49(D) Dis. 2008:13-26.

5. J. Kennedy and R. Eberhart, "Particle Swarm Optimization", in: Proc. IEEE Intl. Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV:1942-1948, 1995.

6. Jianbo Yu, Lifing Xi, Shijin Wang:" An improved Particle Swarm Optimization for evolving Feedforward Artificial Neural Networks"; Neural process let (2007) 26:217-231/ Doi: 10.1007/s 11063-007-90-53-x/ Springer science+ Bussiness media, LLC.2007.

7. M. Dorigo, V. Maniezzo and A. Colorni. "Ant System: optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man and Cybernetics - Part B, vol. 26, no. 1, pp. 29-41, 1996.

8. Marco Dorigo and Mauro Birattari, "Swarm intelligence, (2007) Ant colony optimization. Scholarpedia, 2(3):1461.

9. Matthew Conforth and Yan Meng, "Reinforcement Learning for Neural Networks using Swarm Intelligence", 2008 IEEE Swarm Intelligence Symposium St. Louis MO USA,

10. Pillai , K. G, "Overlapping swarm intelligence with artificial neural network, Swarm intelligence(SIS), 2011 IEEE Symopsium, 15 April 2011. September 21-23, 2008, 978-1-4244-2705-5/08/2008 IEEE.

11. Venu G. Gudise and Ganesh K.V., "Comparision of Particle Swarm Optimization and Back Propogation as training algorithm for Neural Networks"; IEEE Department of electrical and computer engineering, university of Missouri, USA; 0-7803-7914-4/2003 IEEE.

12. W. Eric, Yan Shi, Yu Qi and Richar Golden: "Using an RBF Neural Network to locate program bugs"; 19th International symposium on software reliability engineering; 1071-9458/ 2008 IEEE.

13. Wen-Yu, Zhen-Hai, Xin Liu and Jian-Zhou Wang: " Prediction of ozone concentration using Back Propogation Neural Network with a novel hybrid training algorithm"; 2010 sixth international conference on natural computation (ICNC 2010); 978-1-4244-5961/ 2010 IEEE.

❐