Ant Colony Optimization for Network Aggregation in a Fully Connected Network

Prof. Leena Raut
Department of Computer Technology
Babasaheb Gawde Institute Of Technology
Mumbai
leenakthakur1@gmail.com

Mr. Rutwij Chaudhari
Department of Information Technology
Fr. Conceicao College Of Engineering
Mumbai
rutwijchaudhari@gmail.com

Abstract—Ants Colony exhibit complex structure called ant streets. We were observed in several situations such as highway or may have placed some obstacle in the way and see ant's reaction to such disturbances. One of the most surprising behavioral patterns by ants is the ability to find shortest path which is the challenge of today's computer scientists. Ant Colony Optimization is a probabilistic technique for solving computational problems. This paper explains Ant Colony Optimization is an effective approach for finding a shortest path between two nodes on a network.

Keywords- *Ant Colony optimization, shortest path, probabilistic technique.*

*****

I. INTRODUCTION

Ant Colony Optimization is an algorithm used to find the shortest path in a network from a source to a destination. We are going to build system which can find all the possible paths from the source to the destination and rank these paths to find the shortest path. There are many other algorithms to find the shortest path in a network but we implement this algorithm that is based on the natural metaphor of ant colonies. Most algorithms do not give the user all the possible paths along with their rankings. We give them all the paths as well as allow the user to decide the number of ants and the epochs so that they can have set their own values according to their accuracy needs. We also provide them a system where a nontechnical user can directly load a graph and find the shortest paths.

Economics of this project needs to be understood. Infrastructure requirements would not be a major concern as there is no database required for this project. The major concern would only be the front-end GUI interactions for clients. The project just needs an error-free client side. This project does not need any special requirements. It just needs an IDE or a JDK to run the project. As this project aims at optimization the network paths, it can be best used to compare the various paths for parameters such as duration, cost etc. It can also be used to find the shortest paths for greater efficiency and accuracy.

II. RELATED WORK

In-network data aggregation exploits data correlation and performs aggregation at intermediate nodes to reduce redundant data in network traffic [1]. This leads to reduced amount of communication, end-to-end delay, and network energy. The optimal aggregation tree problem is NP-Hard[2]. Efficient approximation algorithms for the optimal aggregation problem have been studied extensively. The optimization problem is modeled as a combinatorial optimization problem which can be solved using a population based meta-heuristic approach [3]. In addition, energy balanced and efficient network routing protocol is critical to prolonged network lifetime. Ant colony optimization is a meta-heuristic initially proposed by Marco Dorigo in his PhD dissertation in 1992. The original idea comes from observing the exploitation of food resources among ants, in which ants with individually limited cognitive abilities can collectively find the shortest path between a food source and the nest [4]. Ants explore the area surrounding their nest initially in a random manner. As soon as an ant finds a source of food, it evaluates the quantity and quality of the food and carries some to the nest (sink node). During the back tracking, the ant deposits a pheromone trail on the ground. The quality and quantity of the pheromone will guide other ants to the food source [5]. The pheromone trails are simulated via a parameterized probabilistic model. The basic component of an ACO algorithm is "a constructive heuristic that is used for probabilistically constructing solutions using the pheromone parameters" [3]. ACO is first used to solve traveling salesman problem (TSP). Because of the characteristics of distributed computing, self-organization, and positive feedback, it has also been used in prior work for routing in sensor networks [6, 7]. In 2006, Misra and Mandal introduced the ant aggregation algorithm for optimal data aggregation in wireless sensor networks [3]. Many improved algorithms have been proposed afterwards. Those algorithms use a simple heuristic by only considering the distance to the sink node. Liao, Kao, and Fan introduced an algorithm, which makes it easier to seek aggregation nodes [5]. Chen, Guo, Yang, and Zhao proposed an approach using search ants at the beginning for the destination search [8]. Wang and Luo proposed an algorithm composed of path construction, path maintenance, and aggregation schemes including synchronization, loop-free, and collision avoiding schemes [9]. In these algorithms, after the optimal path is discovered, it is used repeatedly. The relaying nodes' energy is depleted quickly. Frequent maintenance of the network and exploration of new paths are needed. Although route discovery overhead can be reduced, the algorithms do not take into consideration the limitations of WSNs, especially energy limitation of sensor nodes and number of agents required to establish the routing [7]. The sensor nodes' lifetimes are highly unbalanced, and consequently the network lifetime is shortened. Multi-path routing algorithms based on ACO have been proposed in recent years. Xia, Wu, and Ni incorporated three new rules in the algorithm to solve the problems of local convergence, local optimization, and multi-path for transferring data, respectively [10]. Xia and Wu proposed an energy-aware multi-path routing algorithm that considers the available power of nodes and the energy consumption of each path as the reliance of routing selection [11]. Yang, Xiong, and Xu proposed a load balancing scheme to distribute the traffic over multi-paths discovered [12]. The network lifetime with these improved algorithms is shown to be longer than the conventional ACO method. However, these routing algorithms do not aggregate data in the routing process. This paper introduces new heuristic rules used in the dynamic-path approach integrated with in-network data aggregation into ACO algorithms in order to extend network lifetime. We compare the conventional and proposed ACO algorithms that utilize different heuristics, node selection rules, and aggregation schemes. The algorithms guarantee packet delivery. Besides, it is shown that the routes discovered are close to the shortest path.

III. PROBLEM STATEMENT

The biggest problem in sending data across network is the searching the possible paths across the network and finding the shortest path amongst them. In real practices, these are essential as it helps improving the energy efficiency and also the congestive routing traffic and reducing the data redundancy.

Thus, there is a need to develop an algorithm that helps to find the shortest most efficient path across any two nodes in a network. Additionally the user might also want to know about the various possible paths and compare them based on the time taken. The user might also want to know the ranking of these paths to select the path according to the parameters he/she wants.

A. *Path Discovery Procedure*

The procedure is composed of the forward and backward passes. The forward pass is a routing process based on the heuristic rule and the pheromone amount on the edges. Backtracking is used in the forward pass when an ant finds a dead end or runs into a loop. In the backward pass, the ant updates the pheromone amount on the path according to the quality of it. The path discovery procedure integrates data aggregation, loop control, and network maintenance. Forward pass. Each ant is assigned a source node. The ant starts from the source node and moves towards the sink node using ad-hoc routing. The aggregation in the forward pass is opportunistic. If multiple ants arrive at the same sensor node simultaneously, aggregation is performed. A path running into loop is prevented by having the ants remember its visited hops. An ant is considered failing its task in an iteration if all the neighboring nodes of the current node have been visited. In that case, the ant changes the heuristic to "choosing the shortest path." A path resulting in "failure" is discouraged in the pheromone deposit process. The forward pass ends when all the ants have arrived at the sink node. Single ant-based solution construction uses the following steps:

1. If the node has been visited in the same iteration, follow a previous ant's path; or
2. Use a node selection rule; or
3. If all the neighbors have been visited, use the shortest path; or
4. If no neighboring node exists, backtrack to the previous node; or
5. If there are no neighbor nodes and the previous node is dead, record the network lifetime and exit the process.
6. Transmit the packet.

B. *Next Node Selection*

Two rules are used for next node selection: LeadingExploration and CombinedRule. The ResidualEnergy algorithm uses LeadingExploration. The SinkDistComb and SinkAggreDist algorithms use CombinedRule so that dynamic paths are established.

1) *LeadingExploration*

Among all the neighboring nodes, select the first node with the highest probability, even if there are multiple nodes with the same probability. An ant always discovers the same optimal path to the sink node until one of the intermediate nodes dies. The advantage of this rule is the current best node is always used. The disadvantage is calculation among all the neighboring nodes and potential repeated use of the same node since pheromone rule biases previously used nodes.

2) *CombinedRule*

Node selection is divided into sessions. Each session includes one or more iterations. A node discovered from the current or a previous iteration is used. Similar to LeadingExploration, the probability of each neighboring node is calculated. A group of nodes with highest probability is stored in a cache. One node is randomly selected and removed from the cache. When the cache is empty, the probability calculation of all the neighboring nodes is repeated. The goal is to avoid recalculation and frequent node reuse in order to balance energy distribution. Meanwhile, the node with the highest probability is always used, so the path selection maintains good quality.

3) *Probability Calculation*

The probability of each neighboring node is calculated using Eq. (1).

$$p_k(i,j) = \frac{\tau(i,j)\times\eta(i,j)^{\beta}}{\sum_{u=0}^{N_i}\tau(i,j)\times\eta(i,j)^{\beta}} \qquad (1)$$

In Eq. (1), an ant K having data packet in node i chooses to move to node j toward the sink node, where τ is the pheromone, η is the heuristic, Ni is the set of neighbors of node i, and β is a parameter which determines the relative importance of pheromone versus distance (β>0). Value η is calculated using Eq. (2). Parameter L represents the total number of heuristic factors, and l is the individual factor. Each factor carries a cost and a weight. The weight is a fraction, and the sum of all the weights equals 1 [13].

$$\eta(i,j) = \sum_{l=0}^{L} Cost_l \times Weight_l. \qquad (2)$$

Three possible costs we consider in Eq. (2) are as follows. Costdist.-to-sink is the inverse of the distance between node j and the sink plus one. Costresidual-energy is the remaining energy of the candidate node. Costdist.-to-aggre.-node is the inverse of the distance between node j and the aggregation node plus one [13].

C. Pheromone Update Rules

The pheromone value is associated with the edge between two nodes. Each edge has pheromone initialized with the same value. The value is updated in the backward pass in order to bias the node selection in the next iteration [13].

1) Evaporation on all edges

After all the ants finish the forward pass and before they are going backward, the pheromone values on all the edges in the network evaporate at rate ρ (Eq. (3)) [13].

$$\tau_{ij} = (1-\rho)\times\tau_{ij}. \qquad (3)$$

IV. PROJECT DESCRIPTION

A. Overview of the project

The various architectural components are :

1.Random NDA : It creates a random network data problem and sets it our panel. It also finds the distances between each node in the network and paints the nodes.

2.Ant Colony : It creates an ant colony with the number of ants and the ACO parameters. It also initializes the ants with the start node and the constraint parameters. It paints the nodes and their edges.

3.User Interface : User Interface provides set of operation to be operated on data. It is an intermediary between the user and the system.

4.Backend Data structures : It is used to store all the possible paths between the source and the destination and local and global best tours.

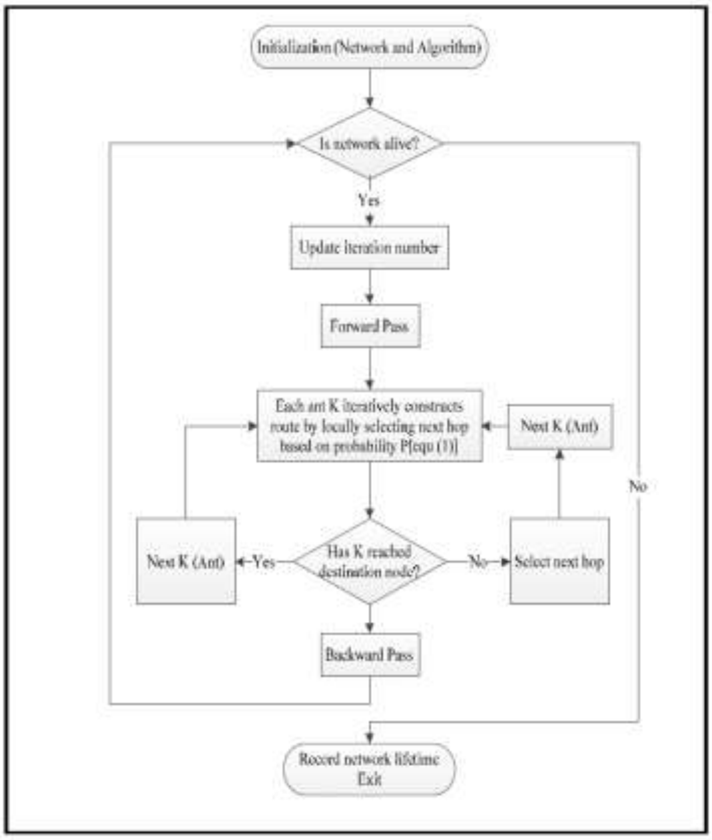
5.User : User provides his input network data problem or loads it directly, the ant colony constraints parameters and enters the source and destination as per his needs.

B. Assumptions

Our System presently works on a fully connected network problem. Which in making is plot of random nodes on x,y axis. The weight of paths between every two nodes is their calculated euclidean distance. Thus when working with partially connected networks the results may differ and increase complexity.

V. IMPLEMENTATION DETAILS

The entire implementation has been divided into five stages. In the first stage, information of Network was collected. In the second stage, information about ant colony was fetched . The third stage involved applying the ACO algorithm on the input data to obtain trail information as data structures. In the next stage, Pseudo-Random-rule is applied to build a solution and local trail update. The final step involves a Global update once all ants complete their solutions.



A. Pseudo Code

Begin:

[1] Initialize Network, Ant Colony and Source and Destination nodes
[2] Calculate weight of every path in the network
[3] Start Timer
[4] Increment, At each step of Timer make a call to run all ants
[5] Choose a random node for the ant to travel from the source node
[6] While Destination not reached Go To Step 5
[7] Calculate the cost of the current path from source to destination
[8] Place local pheromone update
[9] Perform Evaporation
[10] While Number of epoches not completed Go To Step 4
[11] Place global pheromone update
[12] Calculate the best tour and display the results with graphics

End.

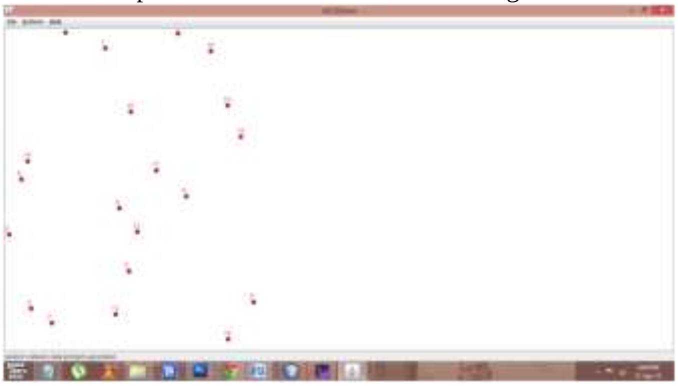
 ISSN: 2321-8169
3873 - 3877

B. *Module Description*

1) MODULE 1: Generate a random NDA

Input : The number of vertices and seed for random generator. As shown in Fig.

Output : It creates a network data problem and sets it to our panel. It also finds the distances between each node in the network and paints these nodes. As shown in Fig.

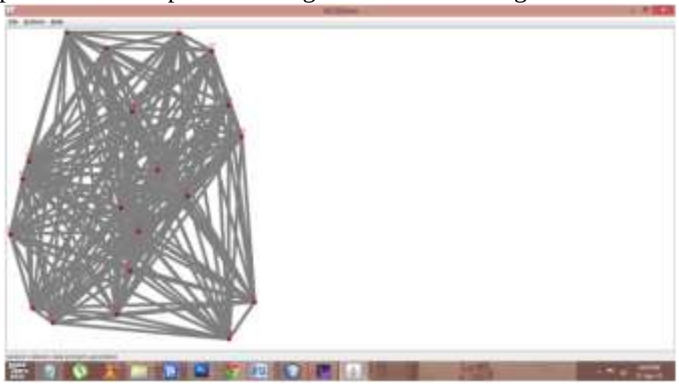


2) MODULE 2 : Create ant colony

Input : The number of ants
The seed for random numbers
The initial pheromone value
The exploitation probability
The pheromone trail weight
The inverse distance weight
The evaporation fraction
The trail laying exponent
The elite enhancement

As shown in Fig.

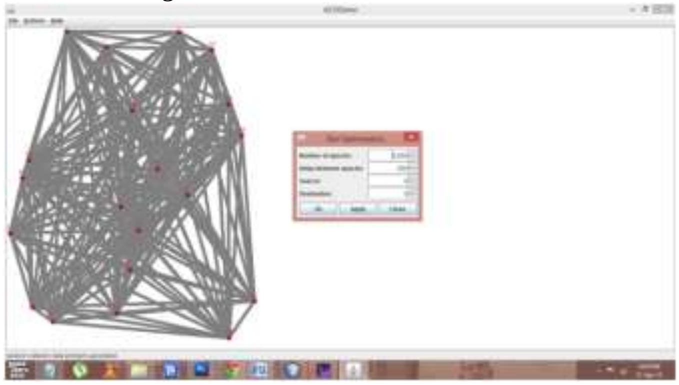
Output : It creates the ant colony along with the parameters. It also initializes the ants with their start node and constraint parameters and paints the edges. As shown in Fig.



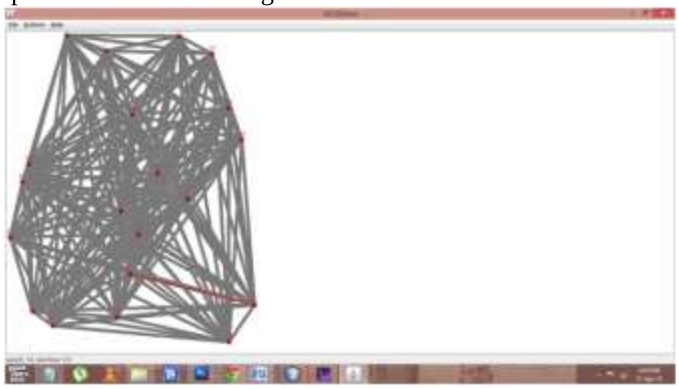
3) Run the optimization

Input : The number of epochs
The delay between epochs
Source node
Destination node

As shown in Fig.



Output : It calls to run all ants and runs every single ant from source to destination through a random path. It then repaints best tour in red and updates status bar with best tour and no of epoches. As shown in Fig.



VI. Results and Future Scope

Our ant colony system currently finds the shortest path in a fully connected network. It does not find shortest paths in a partially connected network. So in the future enhancements we can allow the user to upload or define a partially connected network and then find the shortest path between the source and the destination. Also it does not compare our system with any other shortest path algorithms. So, we can implement other shortest path algorithms like Dijkstra's algorithm, Bellman Ford algorithm, A* search algorithm etc and compare the shortest path in both the algorithm to study the accuracy of our system and also to find the efficiency. We can also include some test cases like varying the number of ants or the number of epochs or both and study the effects of changing these constraints on our outcome and which is the most optimum input for getting an accurate output from our system.

The implementation can also be extended to actual hardware by configuring network devices and hosts to enable real time routing using this algorithm.

VII. Conclusion

Ant Colony Optimization using Network Aggregation is a probabilistic technique which helps us to find all the possible paths from a user specified source to a destination in a fully connected network. It calculates all the possible paths and their distances and ranks them in order to find the shortest path amongst them. It uses pheromone updation as well as Euclidean distances in finding the paths. Therefore it is a very useful system for finding shortest path in a closed network.

VIII. References

[1] G. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, “Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks,” Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pp. 293 – 306, 2006.

[2] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network aggregation techniques for wireless sensor networks: a survey,” IEEE Wireless Communications, vol. 14(2), pp. 70 – 87, 2007.

[3] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, "Data aggregation in wireless sensor networks - exact and approximate algorithms," High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on, pp. 241- 245, 2004 82

[4] R. Misra and C. Mandal, “Ant-aggregation: ant colony algorithm for optimal data aggregation in wireless sensor networks,” Proceedings of IFIP International Conference Wireless and Optical Communications Networks, vol. 5, 2006.

[5] “Ant colony optimization algorithms,” 2010. Retrieved Nov 6th, 2011, from http://en.wikipedia.org/wiki/Ant_colony_optimization

[6] W. Liao, Y. Kao, and C. Fan, “An ant colony algorithm for data aggregation in wireless sensor networks.” International Conference on Sensor Technologies and Applications, pp. 101 – 106, 2007.

[7] S. Das, G. Singh, S. Gosavi, and S. Pujar, “Ant colony algorithm for data-centric routing in sensor networks,” Proceedings of the Joint Conference on Information Sciences, pp. 1565 – 1568, 2003.

[8] J. Yang, Y. Lin, W. Xiong, and B. Xu, “Ant colony-based multi-path routing algorithm for wireless sensor networks,” Proceedings of the International Workshop on Intelligent Systems and Applications, pp. 1 – 4, 2009.

[9] G. Chen, T. Guo, W. Yang, and T. Zhao, “An improved ant-based routing protocol in Wireless Sensor Networks” Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 1 – 7, 2006.

[10] H. Wang and N. Luo, “An improved ant-based algorithm for data aggregation in wireless sensor networks. Proceedings of the 2010 International Conference on Communications and Mobile Computing, vol. 3, pp. 239 – 243, 2010.

[11] S. Xia, S. Wu, and J. Ni, “A new energy-efficient routing algorithm based on ant colony system for wireless sensor networks,” Proceedings of the 4th International Conference on Internet Computing for Science and Engineering, pp. 176 – 180, 2009.

[12] S. Xia and S. Wu, “Ant colony-based energy-aware multipath routing algorithm for wireless sensor networks,” Proceedings of the 2nd International Symposium on Knowledge Acquisition and Modeling, pp. 198 – 201, 2009.Article in a journal:

[13] Meng Xie, “Ant Colony Optimization Based In Network Data Aggregation In Wireless Sensor Networks” International Symposium on Pervasive Systems, Algorithms and Networks, pp. 79 – 80, 2012