Nature Inspired Approach for Path Determination Using NetLogo

T.R. Gopalakrishnan Nair[1], Kavitha Sooda[2] , Prabhjot Singh[3] , Nikhilesh Vaishy[3] , Aayush Vats[3] , Danish Iqbal[3]

[1]*Saudi ARAMCO Endowed Chair – Technology, PMU, KSA Advanced Networking Research Group VP, Research, Research and Industry Incubation Center (RIIC), Dayananda Sagar Institutions, Bangalore 560078, India*
e-mail: trgnair@ieee.org, www.trgnair.org

[2]*Advanced Networking Research Group (RIIC), DSI and Assoc. Professor, Dept. of CSE*
Nitte Meenakshi Institute of Technology
Bangalore 560064, India
e-mail: kavithasooda@gmail.com

[3]*Dept. of Computer Science & Engineering*
Nitte Meenakshi Institute of Technology
Bangalore 560064, India
e-mail: prabhu0376@gmail.com; nikhilesh.vaishy@yahoo.in; aayushvats101@gmail.com; syeddanishiqba89@gmail.com

Abstract**—** In the current internet routing scenario, it is very important that data packets on any network reach the destination quickly with cognition approach. This is required so as to reduce congestion in the environment even before routing can take place. Here we apply the network routing approach using Ant Colony Optimization (ACO). ACO has been used as a search process for optimality exploitation and exploration. The paper shows how ACO approach has been utilized for determining the optimal path based on bandwidth availability of the link as well as distance between the nodes with a comparison against Dijkstra's algorithm. Here pheromone can be considered analogous to bandwidth. An agent system governs the collection of QoS parameters of the nodes. The simulation was carried out in Netlogo and the result shows that the ACO explores a large number of possible paths to destination and converges to the most optimal path with respect to Dijkstra's algorithm which just considers the shortest path to destination.

Keywords—ACO; Routing; Optimal path; MAS; Pheromone; NetLogo; QOS

I. INTRODUCTION

Technological advancement has made the people want faster connections between their machines connected on a network. Some techniques proposed to get such connections are successful while several others are not. There is no efficient algorithm yet for the internet as a whole. Therefore, future network algorithms should be robust and work in a distributed way. In future, Artificial Intelligence may be used to control flow of data packets on a global level.

One of the several techniques proposed for efficient routing involves study of ant behavior while finding fastest path between their nest and food. Computer scientists have been researching ant behavior implementation in networking applications since 1990's. Ant Colony Optimization (ACO) algorithm is one of the outcomes of these research works. The use of several simple biological agents (for e.g. the ants, bees etc) generates a variety of organized behaviorally patterns at system-level as a result of local interactions among the agents and the environment. [1].

There are many advantages of ACO algorithm application in routing that can be compared to real ants:

- The ants or the agents select the best path among all the available paths.
- The data packet which has to reach destination is ensured. Because, when an ant is presented with an obstacle, it selects a new best path.

Ants possess the capability to create bidirectional routes between their home and their goals that is their food source in optimal time. These paths may develop obstructions which cannot be crossed and the path may need to be changed at any point in time. It is this ability of the ants which interests researchers the most. By mimicking these abilities with routing algorithms researchers believe they can help the internet as a whole grow faster.

This paper presents the implementation of ACO on Netlogo which shows ants behavior in finding the optimal path in a network using the multiagent systems (MAS) behaviour.

The rest of this paper is organized as follows; in Section 2 related literature is described and in the next section ACO is discussed in detail. The results are discussed in section 4 and conclusion is provided in section 5.

II. RELATED WORKS

Several algorithms have been proposed, implemented and analyzed for solving shortest path problem in [7][8]. Although these algorithms are simple and easy to use, they have several drawbacks which make them inappropriate for dynamically changing environment. In these cases, the routing decisions are merely based on the table information, provided at the node level. Such type of scenario has no real awareness of the surrounding environment. This is where ACO based routing can play its role through the intelligence aspect incorporated in it. Many researchers are working in the area of ACO since a decade. Many nature inspired techniques have been developed and explored over the years including Artificial Bee colony algorithm (ABC), Glowworm swarm optimization etc.

Artificial bee colony algorithm (ABC) was introduced by Karaboga in 2005 [9], and simulates the foraging behaviour of honey bees. Its working algorithm has three phases: employed bee, onlooker bee and scout bee. In the employed bee and the onlooker bee phases, bees exploit the sources by local searches in the neighborhood selected based on deterministic selection in the employed bee phase and the probabilistic selection in the onlooker bee phase. In the scout bee phase which is similar to abandoning exhausted food sources in the foraging process, non-beneficial solutions are abandoned and new solutions are looked to explore new regions in the search space.

In Glowworm swarm optimizations agents are glowworms that carry a luminescent quantity called luciferin along with them. The luceferin value is encoded as a fitness function and is then broadcasted to the neighbours in search of the optimal solution.

III. ANT COLONY OPTIMIZATION AND MULTIAGENT SYSTEM

A. *Multi-Agent Systems*

Agent-based systems technology has been hailed as a new paradigm for conceptualizing, designing, and implementing software systems. As the numbers of complex problems are increasing, we need a sophisticated computer program that can act autonomously. Agents possess the ability to act autonomously on behalf of their respective users across a large distributed and open environment [3]. Increasingly, however, applications require multiple agents that can work together independently. An agent is a computer system that can take independent actions on behalf of their user or owner and are deployed in a dynamic environment [4].

When a number of agents interact with each other, it is known as multi-agent system (MAS). They solve problems that are beyond individual capacities or knowledge of each problem solver. With different goals and motivation, agents act on behalf of users in multi-agent system. There are certain characteristics described in [2] that an agent requires for successful interaction: cooperate, coordinate, and negotiate with each other. Multi-agent systems are made up of multiple interacting intelligent agents—computational entities to some degree autonomous and able to cooperate, compete, communicate, act flexibly, and exercise control over their behavior within the frame of their objectives [2].

B. *Ant Colony Optimization*

Ants, in nature, exhibit some complex social behaviors that have attracted the attention of many human beings. Ant colonies, and more generally social insect societies, are distributed systems that, in spite of the simplicity of their individuals, present a highly structured social organization [5]. As a result of this organization, ant colonies can accomplish complex tasks that in some cases far exceed the individual capacity of a single ant.

The visual perceptive faculty of many ant species is only rudimentarily developed and there are ant species that are completely blind. One of the most important insights of the early research on ants was the communication between individuals, which is based on the chemicals produced by ants [6]. This chemical is known as pheromone. This is different from other species for example, humans and other species, whose important senses are visual or acoustics. Some of the species of the ants leave a pheromone trail. By sensing this pheromone trail, other ants that are foraging, follow this path to the food source that other ants have explored. This collective trail-laying and trail-following behavior of ant is influenced by the chemical trail left other ants, which is the inspiring source of ACO.

In ACO it is the regular ants which transport packets from their origin to destination in the most efficient manner. At every node in the path these ants look at the probabilistic routing tables, which have been set up initially by the uniform ants, and determine which next hop to take as it progresses to the destination. When a route is taken more often and is proved to be the most efficient route, its probability of being taken again will be increased. This will make all the regular ants converge on to that route which is then taken by almost all of the packets. When this state is achieved the ants are said to be stable. This is how the technique finds the fastest path through the topology. The regular ants do not use as much intelligence as the uniform ants, which are used to discover the fastest routes through the network.

$$P(e) = \frac{\tau(e).\eta(e)}{\Sigma_{available\ edges\ e'}\ \tau(e').\eta(e')} \qquad (1)$$

In (1), *P(e)* is the probability of choosing the edge *e*, from the list of available edges *e'*. If we consider a network graph as the natural habitat of ants, then attractiveness,τ for an edge in the graph is the pheromone level on that edge. The parameter for edge selection in our work is bandwidth, which is analogous to pheromone level on each edge. The specific visibility function (η), is the reciprocal of distance between two nodes.

An inherent trait in networking which affects ACO and other routing algorithms is that bad news travels fast and good news travels slow. If a router suddenly goes down the ants are trained to deal with this situation and do so quickly. As the reports of ants reaching their destination stop coming back to the router, the next ants will choose the second best path and the best route will be altered quickly. However, if for some reason the best route which all the ants choose slows down or for instance if that router has a high load transferring through it, then the algorithm does not go to the second best route as quickly.

There are several disadvantages as well while applying ACO in network routing:

- The time taken by ACO to converge to the destination is much more than that of Dijkstra's Algorithm. This is because of the exploration of path in ACO.
- It is also difficult to implement ACO with network characteristics such as traffic load, network topology etc.
- In dynamic network routing problems, need solutions for changing conditions focus is on effective evaluation of alternative path.

Few other applications of ACO include:

- **Travelling salesman problem**: It asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"
- **Quadratic assignment problem** : The problem models the following real-life problem , "There are a set of *n* facilities and a set of *n* locations. For each pair of locations, a *distance* is specified and for each pair of facilities a *weight* or *flow* is specified (e.g., the amount of supplies transported between the two facilities). The problem is to assign all facilities to different locations with the goal of minimizing the sum of the distances multiplied by the corresponding flows.
- **Vehicle routing**: The problem seeks to service a number of customers with a fleet of vehicles. The objective of vehicle routing is to minimize the total route cost.
- **Graph coloring**: It is a way of coloring the vertices/edges of a graph such that no two adjacent vertices/edges share the same color.
- **Sequential ordering problems:** Involves finding a minimum cost Hamiltonian path on a directed graph with costs associated to arcs and nodes.

C. *Implementation*

Fig. 1 depicts the series of steps needed to implement the problem . For optimal path selection needed in effective data routing, we have chosen the Ant Colony Optimization (ACO) algorithm which is one of the promising techniques for handling complex problems. The algorithm implementation as well as the network model simulation has been done using NetLogo, a multi-agent based modeling environment.

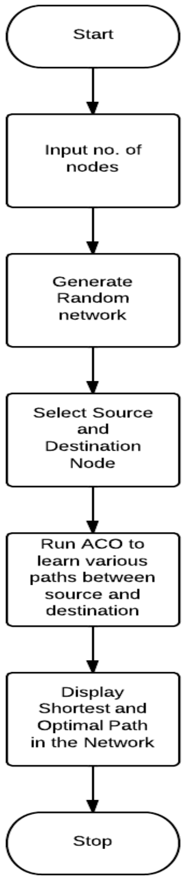

Fig. 1. Implementation Flow Chart of ACO

IV. RESULTS

The simulation is carried out in NetLogo v5.0.5 which is a multi-agent programming environment. It is based on the language called Logo which is used for functional programming and to generate shapes with its basic agents. There are four types of agent in NetLogo: Turtles, Patches, Links and Observer. In the NetLogo world, Turtles are agents that move around in the world. This world can be two dimensional or three dimensional and is divided into a grid of patches. The turtles can move on patch which is a square piece of *ground*. Each turtle has a set of properties associated with it like x-coordinate, y-coordinate, color, heading, id etc. Links can be defined as those agents who connect the turtles. The observer looks out over the world of turtles and patches, and gives instruction to them – it doesn't observe passively. The interaction between these agents simulates the real world behavior of various kinds of phenomena.

A network is generated using all the agents with assigned properties and the links are assigned bandwidth. Two nodes are selected randomly as source and destination and the algorithm attempts to find the optimal path between them.

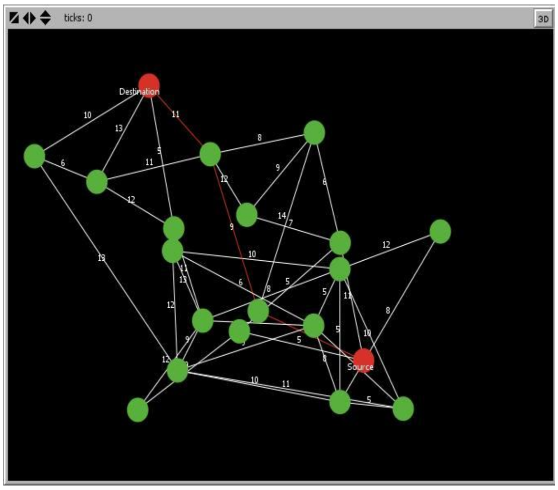

Fig. 2. Randomly generated network in NetLogo

In the network topology as depicted in Fig. 2 which has been generated and simulated on NetLogo, we can see a clear source and destination. The respective bandwidth of the links is displayed over the links connecting the nodes. In our work, we have considered bandwidth as our primary edge selection parameter while ACO algorithm is being implemented.

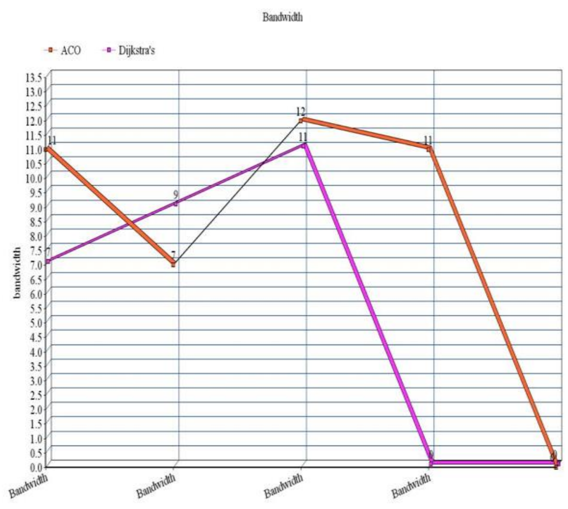

Fig. 3. Hop-by-hop bandwidth comparison for the two algorithms

A result was obtained which can be seen in Fig. 3 that ACO's edge selection was more intelligent than Dijkstra. While given a choice to select a from among a number of edges originating from the same node, ACO implementation made sure that the edge with the highest bandwidth gets selected. Upon implementing both ACO and Dijkstra's Algorithm, on the same network topology one after the other in Fig. 4, we have determined that even though Dijkstra's algorithm took lesser time to converge to the destination, it was found out that ACO explored the network a lot better to find the optimal path in the network.

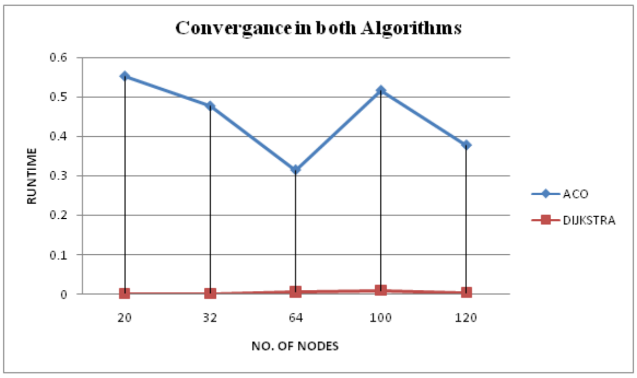

Fig. 4. Convergence time comparison

Table I shows a relative comparison of both the algorithms, ACO and Djikstra's in terms of the number of hops taken on the best path for a network with the given number of nodes.

TABLE I. Comparison of Hop Counts of Best Path Obtained

No. of Nodes	Dijkstra's	ACO
20	2	4
32	3	4
64	3	3
100	7	5
128	5	5

The results show that ACO tends to take more number of hops during exploration compared to Djikstra's for a given network with the same number of nodes. As the number of nodes increases ACO takes more number of hops when compared to Djikstra's. For instance, for a node count of 20, Djikstra's and ACO take 2 and 4 hops respectively whereas the same take 3 and 4 hops respectively for a node count of 32. Clearly this is attributed to the exploratory nature of the ACO algorithm that makes it take slightly higher number of hops with respect to Djikstra's.

Considering all the parameters discussed above, ACO collectively performs a better task in exploring the optimal path when compared to Djikstra's. It also shows all the possible optimal paths that exist between the source and the destination and then selects the most optimal one.

V. CONCLUSIONS

In this paper we have successfully implemented the concept of ant colony optimization (ACO) on NetLogo. We have also compared this algorithm with Dijkstra's Algorithm for routing data packets. Results show that although Dijkstra's Algorithm converges faster from the source to destination, but ACO does a better job in exploring the network and gives you more number of possible paths between the two locations while converging to an optimal path. The ACO selects the next edge intelligently based on the parameter of bandwidth of all links from that node, that is the link with the highest bandwidth gets selected based on certain probabilistic fluctuations.

REFERENCES

[1] Mrs. B.D. Shirodkar, Dr. S.S.Manvi, A.J.Umbarkar, "Multicast Routing for Mobile Ad-Hoc Networks using Swarm Intelligence", International Journal of Recent Trends in Engineering, Pages: 36-40, Vol 1, No. 1, May 2009

[2] Gerhard Weiss: "Multiagent Systems: A modern approach to distributed computing", The MIT Press, 2nd Edition, 2013

[3] Li Zhengping, Malcolm Yoke Hean Low, Arun Kumar, "A Framework for Multi-Agent System-Based Dynamic Supply Chain Coordination", Multi-Agent System-Based Logistics Coordination

[4] Michael Wooldridge:"An Introduction to MAS", George Wiley & Sons, 2nd Edition, 2009

[5] Radoslav Andreev, Patrick Healy, Nikola S. Nikolov, "Applying Ant Colony Optimization Metaheuristic to the DAG Layering Problem" Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International, Pages: 1-9, March 2007

[6] Marco Dorigo and Thomas Stutzle: "Ant Colony Optimization", 1st Edition 2004

[7] Kavitha Sooda, T. R. Gopalakrishnan Nair: Optimal path selection in graded network using Artificial Bee Colony algorithm with agent enabled information, HIS 2012: Pages: 330-335

[8] E.W. Dijkstra, A Note on Two Problems in Connexion with Graphs, Numerische Mathematik Volume 1, Issue 1, Page: 269-271, 1959

[9] Karboga , Devis , " Artificial Bee Colony Algorithm " , Scholarpedia 5 (3): 6915, 2010

T.R.Gopalakrishnan Nair holds M.Tech. (IISc, Bangalore) and Ph.D. degree in Computer Science. He has 3 decades experience in Computer Science and Engineering through research, industry and education. He has published several papers and holds patents in multi domains. He is winner of PARAM Award for technology innovation. Currently he is the Saudi Aramco Endowed Chair, Technology and Information Management, PMU, KSA and VP of Research and Industry in Dayananda Sagar Institutions, Bangalore, (Sabbatical), India.

Kavitha Sooda holds M.Tech in Computer Scinece and Engineering. She has eleven years of teaching experience and pursuing her Ph. D from JNTU, Hyderbad. Her interest includes Routing techniques, QoS application, Cognitive networks and Evolutionary algorithms. She has twelve papers based on her research area. Currently she works as Assistant Professor at Nitte Meenakshi Institute of Technology, Bangalore, India.

Aayush Vats just completed his B.E. in Computer Science and Engineering from Nitte Meenakshi Institute of Technology, Bangalore. His areas of interest include Networking and Internet Security.

Nikhilesh Vaishya just completed his B.E. in Computer Science and Engineering from Nitte Meenakshi Institute of Technology, Bangalore. His areas of interest include Networking and Internet Security.

Prabhjot Singh just completed his B.E. in Computer Science and Engineering from Nitte Meenakshi Institute of Technology, Bangalore. His areas of interest include Networking and Cryptography.

Danish Iqbal just completed his B.E. in Computer Science and Engineering from Nitte Meenakshi Institute of Technology, Bangalore. His areas of interest include Networking and Finite Automata.