ISSN No. 0976-5697

Volume 5, No. 8, Nov-Dec 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Estimation of Effect of using ACO in Dynamic Routing on a Communication Network

[1]Prof.Sandeep Singh,[2]Sahil Arora,[3]Mohak Saini,[4]Ishaan Arora
[1]Electronic and Communication, [2,3,4] Information Technology,
Maharaja Surajmal Institute of Technology, GGSIPU, Delhi, India

Abstract - Although an ant is a small and simple creature, but collection of ants or a a colony of ants performs useful tasks such as finding the shortest path to a food source and sharing this information with other ants by depositing pheromone. In the field of ant colony optimization (ACO), models of collective intelligence of ants are transformed into useful optimization techniques that find applications in computer networking. In this paper we present an implementation of Artificial Intelligence on any communication network and compare the results thus produced with the traditional routing algorithm like the shortest node first. The problem of routing and congestion are of utmost concern for the design and implementation of any communication network. Here in this paper we present an approach of performing routing with automatic congestion control and loop removal using artificial intelligence. For the purpose of demonstrating the results of our findings we have designed a simulation of a communication network. We also performed a search space optimization process in order to find out the most appropriate algorithm to be implemented. The comparison and analysis of AI and Non AI modes is performed and is displayed in terms of different graphs. The proposed implementation of AI techniques in routing and congestion control provides a better solution than the traditionally available methods. The algorithm used for dynamic routing is ACO (Ant Colony Optimization) Algorithm which is a metaheuristic algorithm belonging to the class of Swarm Intelligence Algorithms.

Keywords – ACO, Ants, Networks, Routing, and Swarm Intelligence.

I. INTRODUCTION

With the growing importance of telecommunication more complex networked systems are being designed and produced. The challenge of competing with numerous complexities of networking problems such as load balancing, routing and congestion control produce the need for more sophisticated (and perhaps more intelligent) techniques to solve these issues. Working on some of the computing methods inspired by social insects such as ants, several mo-bile agent-based paradigms were designed to solve control and routing problems in telecommunication and networking. Al-though by itself, an *ant* is a simple and unsophisticated creature, collectively a colony of ants can perform useful tasks such as building nests, and *foraging* (searching for food). What is interesting is that ants are able to discover the shortest path to a food source and to share that information with other ants via stigmergy. Stigmergy is basically a form of indirect communication used by ants in nature to coordinate their problem-solving activities. Ants achieve stigmergic communication by laying a chemical substance called *pheromone* that induces changes in the environment which can be sensed by other ants [1].

In recent years, computer scientists were able to transform the models of *collective intelligence* of ants into useful optimization and control algorithms. In this new field of *ant colony optimization* (ACO), a colony of (biological) ants is typically modelled as a society of mobile agents (or artificial ants). Although ACO has been applied in many combinatorial optimization problems such as the asymmetric travelling salesman problem, graph colouring problem and vehicle routing problem, this manuscript focuses on surveying ACO approaches in network routing and load-balancing. In applying ACO in network routing and load-balancing, an artificial ant is typically realized as a simple program consisting of simple procedures that simulate the laying and sensing of pheromone, and data structures that record trip times and the nodes that it passes.

Migrating from node to node, an artificial ant emulates laying of pheromone by updating the corresponding entry in the routing (or pheromone) table in a node which records, for example, the other nodes that this node is directly connected. While a more detailed exposition of the problem-solving paradigm of ACO is given using an example later, the differences between ACO and traditional routing algorithms are discussed later.

II. ANT COLONY OPTIMIZATION

This section describes the problem solving paradigm of ACO in finding an optimal path. Suppose that there are four ants and two routes leading to a food source: R1 and R2 such that R1>R2. Along the two routes, there are six nodes: N (Nest), N1, N2, N3, N4 and F (Food Source). Initially all ants (A1, A2, A3, A4) are at the decision point N and they have to select between R1 and R2 to reach F.

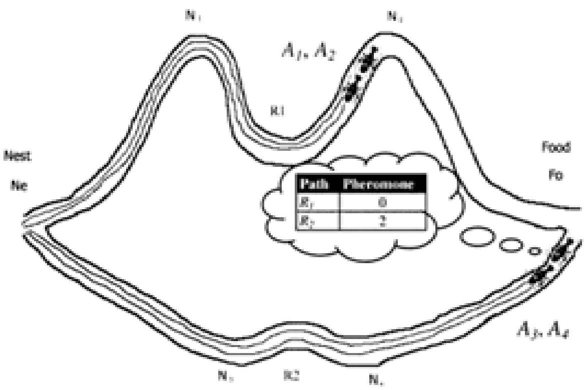

Figure 13 Problem Solving of ants

a. At N, all ants have no knowledge about the location of food. Hence they randomly select from {R1, R2}. Suppose that A1 and A2 choose R1, and A3 and A4 choose R2.
b. As A1 and A2 move along R1 and A3 and A4 move along R2 they leave certain amount of pheromone along their paths _R1 and _R2, respectively.
c. Since R2<R1 A3 and A4 reach F before A1 and A2. When A3 and A4 pass R2 to reach F, _R2=2, but A1 and A2 have yet to reach F and _R1=0. To return to N from F, A3 and A4 have to choose between R1 and R2. At F, A3 and A3 and A4 detects that _R2>_R1, hence they are more likely to select R2. Suppose A3 and A4 select R2.
d. As A3 and A4 pass R2 for the second time to reach N, _R2 is incremented to 4. The increase in _R2 is further consolidates R2 as the shorter path. When A1 and A2 reach F, _R2=4 and _R1=2. Hence, A1 and A2 are more likely to select R2 to return to N. In this example any ant at F (respectively N) will be able to determine the optimal path once A3 and A4 reach F (respectively N). If an ant is at a choice point where there is no pheromone, it makes a random decision with probability 0.5 of choosing R1 or R2. However then

The example in Fig. 1 is an illustration. Here in this figure the model adopted is one in which the pheromone is laid both during the trip to and from the point F. However in *privileged pheromone* laying the pheromone is laid only during the return trip.

III. ACO VERSUS TRADITIONAL ROUTING

In this section, the differences between ACO routing and traditional routing algorithms such as the distance vector routing or link state routing are discussed [2]. Of particular interest are the issues of

A. routing information;
B. routing overhead;
C. adaptivity and stagnation.

A. Routing Information:

In traditional routing algorithms, a node depends on the routing information furnished by all its neighbouring nodes to construct a complete routing table. Furthermore, the neighbouring nodes of any node Ni in turn depend on the routing information of their neighbouring nodes which in turn depend on other neighbouring nodes.

In ACO, the paths from a source to a destination are explored independently and parallely. As and when an ant arrives at a node, then and there the corresponding pheromone value for a path is updated; hence, each entry of a pheromone table in a node is updated independently. This node can immediately use the information in its pheromone table to route data packets to new node depending on the updated information.

B. Routing Overhead:

Traditional routing involves the transmission of routing tables of each node N, to every one of the neighbours. For a large network, the routing table of each of the nodes consist of cost vectors to all other nodes Ni, which is quite large. Since each Ni needs to transmit its routing table to every one of its neighbours, this routing overhead can be very large. Routing in ACO is achieved by transmitting ants rather than routing tables or by flooding LSPs. Even though it is noted that the size of an ant may vary in different systems/implementations, depending on their functions and application, but actually , the size of ants is comparitively small, in the order of 6 bytes and thus the overhead is quite small.

C. Adaptivity and Stagnation:

In dynamic networks, transmitting large routing table or flooding multiple copies of LSPs in short or regular intervals may incur large routing overhead. However, flooding LSPs and transmitting routing table in longer intervals may result in slower responses to changes in network topology. Since ants are relatively small they can be piggybacked in data packets, and more repetitive transmission of ants provide updates of routing information . Hence, the use of ACO for routing in dynamic network seems to be appropriate [6].

TABLE I
ACO ALGORITHMS VERSUS TRADITIONAL ROUTING ALGORITHMS

	RIP / OSPF	ACO algorithms
Routing preference	Based on transmission time / delay	Based on pheromone concentration
Exchange of routing information	Routing information and data packet transmitted separately	Can be piggybacked in data packets
Adapting to topology change	Transmit routing table or Flood LSPs at regular intervals	Frequent Transmission of ants
Routing overhead	High	Low
Routing update	Update entire routing table	Update an entry in a pheromone table independently

Table 3. ACO vs RIP

Related to the issue of adaptivity is stagnation. Stagnation occurs when a network reaches its convergence (or equilibrium state); an optimal path is chosen by all ants and this recursively increases an ant's preference for **P** (optimal path). This lead to:

a. congestion of **P**,
b. dramatic reduction of the probability of selecting other paths. The two are undesirable for a dynamic network since:
a. **P** may become non-optimal if it is congested;
b. **P** may be disconnected due to network failure;
c. The remaining non-optimal paths may become optimal because of the changes in network topology, and iv) new or better paths may be discovered.

These limitations of stagnation are removed by using the techniques of:

a. Evaporation
b. Ageing
c. Pheromone Smoothing
d. Pheromone Heuristic Control
e. Privileged Pheromone Laying

IV. NETWORK SIMULATION

For the purpose of implementation a bi-directional, unweighted topological network consisting of 30 nodes has been created and closely resembles any communication network. After a basic number of parameters have been set the simulation is made to run. Firstly all the pheromone tables are set to default having equal weights and then calls are generated and placed on network. Initially the routes selected are random. If a call cannot connect to node it is forced to wait and the wait counter is enumerated to reflect the quantum (in timer ticks). Once a node has reached its final or destination node it will work its way backwards altering the local nodes pheromone table as it traverses. The shorter the route the greater is the increase in probability given to its table entry in the pheromone table. This happens again and again until the weight of the fastest node is shifted such that slower routes have a very low probability of being chosen [3].

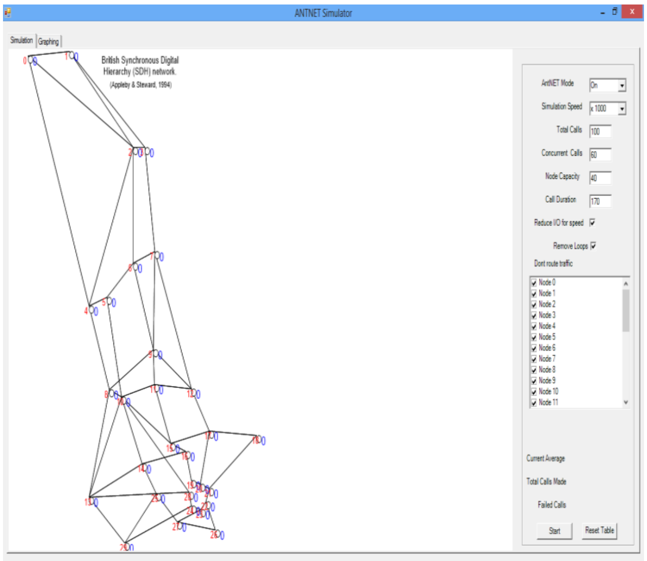

Figure 14 Main Application

V. COMPUTATIONAL DETAILS

General ACO algorithm that we used for our simulation uses following:

A. *ACO Pseudo code:*

Set parameters; initialize pheromone trails
SCHEDULE_ACTIVITIES
ConstructAntSolutions
UpdatePheromones
END_SCHEDULE_ACTIVITIES

B. *Edge Selection:*

An ant will move from the node *i* to the node *j* with a probability

$$\rho(i,j) = \frac{\tau(\alpha)\varphi(\beta)}{\sum\tau(\alpha)\varphi(\beta)}$$

$\tau(i,j)$ is the amount of pheromone on edge *i,j*
α is a parameter to control the influence of $\tau i,j$
φ *i,j*, is the desirability of edge *i,j* (a prior knowledge, like 1 / *di,j*, where d is the distance)
β is a parameter to control the influence of _*i,j*

C. *Pheromone Update:*

$\tau\ i,j = (1 - \rho)\ \tau i,j + \Delta\ \tau i,j$

Where $\tau i,j$ is the amount of pheromone on a given edge *i,j*, ρ is the rate of pheromone evaporation and $\Delta\ \tau i,j$ is the amount of pheromone deposited, typically given by [4]

$$\Delta\tau(i,j) = \begin{cases} \frac{1}{L(k)} & if\ ant\ k\ travels\ on\ i,j \\ 0 & \end{cases}$$

Where *Lk* is the cost of the *k*th ant's tour (typically length).

VI. ROUTING MECHANISM

To begin with, each possible path has an even likelihood of getting chosen. The ant is placed on a network of 4 nodes with the source node of 1 and destination node 2. A chance mechanism is invoked and a path is chosen.

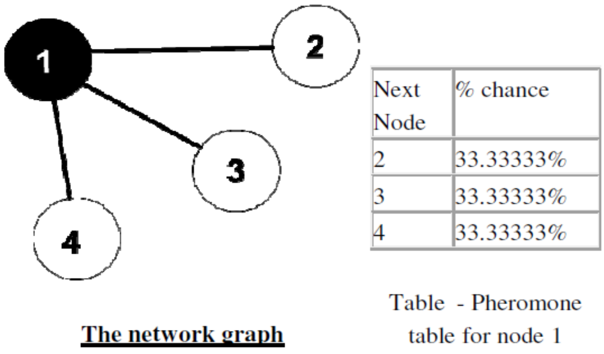

Figure 15: Network graph and table

In this case node 2 has been selected and the ant arrives at its source destination. The ant then moves and updates the pheromone tables for the visited nodes with higher (and more mathematically biased) values. This would be founded for figure 3.2 and table 3.2 in the following way [7]:

a. Node 2 was the final destination
b. It took 1 hop to reach its destination
c. Divide 1 (hop) by 100 : 100%
d. Add 100 to the probability value of node 2 (currently 33.3333): 133.3333
e. Add the values of the other nodes to 133.3333 (133.3333+ 33.3333 + 33.3333): 200 (approximately)
f. Calculate the ratio: ratio = 100/200 0.5
g. Fix the probability of the node to its current value multiplied by the ratio
h. Node 2: 133.3333 * ratio (0.5) = 66.6666%
i. Node 3: 33.3333 * ratio (0.5) = 16.6666%
j. Node 4: 33.3333 * ratio (0.5) = 16.6666%
k. Node 2 (66.6666%) + Node 3 (16.6666%) + Node 4 (16.6666%) = 99.9999%

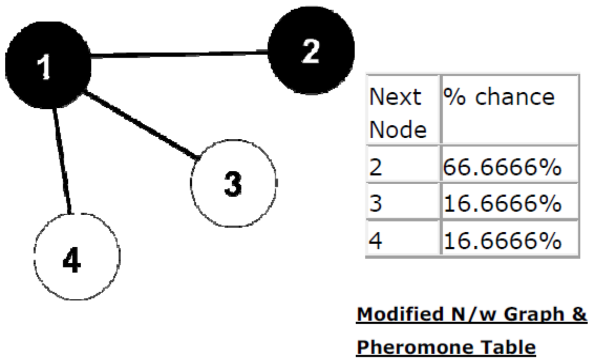

Figure 16: Modified graph and table

In our project we have devised a pheromone base model for determination of the next node from any given present node.

a. Present Node (i.e. Pheromone table for given node)
b. Destination Node
c. Next Possible Node
d. Fitness Value

[Here the term fitness value determines the next node to be selected from the given present node. Fitness value is calculated as fraction of pheromone on this path to the total path multiplied by 100. This gives the percentage probability of the present path to be chosen [5]. The typical structure of any node's pheromone table looks like:

Table 4: Pheromone probability table

Phermomone Tables for Node 4			
	Destination Node	Next Possible Node	Proablity of Node being quickest
	Node 0	Node 0	47.8772587597954%
	Node 0	Node 2	50.9249164080519%
	Node 0	Node 5	0.598912416076378%
	Node 0	Node 8	0.598912416076378%
	Destination Node	Next Possible Node	Proablity of Node being quickest
	Node 1	Node 0	41.5951447777531%
	Node 1	Node 2	58.3095807437051%
	Node 1	Node 5	0.0476372392708999%
	Node 1	Node 8	0.0476372392708999%
	Destination Node	Next Possible Node	Proablity of Node being quickest
	Node 2	Node 0	8.77589861372847%
	Node 2	Node 2	91.1473549041918%
	Node 2	Node 5	0.0383732410398749%
	Node 2	Node 8	0.0383732410398749%
	Destination Node	Next Possible Node	Proablity of Node being quickest
	Node 3	Node 0	9.75196678321678%
	Node 3	Node 2	70.7440996503496%
	Node 3	Node 5	9.75196678321678%
	Node 3	Node 8	9.75196678321678%

VII. ANALYSIS

In order to determine the effect of using the ACO algorithm on Dynamic routing the following type of simulations are performed with varying network and algorithm parameters.

The graphs thus obtained give a general quantitative measure of the effect of ACO in Routing.

A. ACO vs. Non ACO Mode:

To analyse the result of using the AI method (ACO) on a communication network it is compared to a Non-Ai mode. For that the simulation is run for both ACO and Non ACO modes with same parameters for network. The results thus obtained are analysed graphically analysed. The use of ACO generally results in a decrease in the average number of hops. First curve denotes simulation without ACO and the other one denotes simulation with ACO.

B. Insertion of ACO in between a Non ACO Mode:

To view the algorithm from a different perspective the following procedure is followed: first the simulation is run with the ACO algorithm off and then in the next run of the simulation ACO is activated on the 500th call. This can be identified by a label and follows with a decline of average hops. The simulation is run for different number of calls and with different parameters and it is always observed that turning ON the ACO always results in a decrease in the average number of hops. Simulation curve denotes simulation without ACO and the other curve denotes simulation with ACO turned ON at 500th call.

C. Adaptivity Curve:

In order to observe the result of failure of some nodes or routers in between (which is a very obvious error that can occur in the communication networks). We have simulated the same situation in our algorithm as well. For an ACO mode network to be successful it should always work in the case of failure or removal of some of the nodes. Again by running the simulation a number of time we obtained that the ACO handles Adaptivity quite well. Even in case of failure of some of the nodes the ACO algorithm performs as expected and decreases the average number of hops traversed in the call completion.

D. Loop Removal:

Before an Ant returns back to its source node, an optimization technique of loop elimination can be invoked. The issue amongst loops is that they can get several times, the amount of pheromone than they should lead to the problem of self reinforcing loops. In order to *prevent such self enforcing loops we have taken some precautions and in order to check the validity of the methods we adopt to prevent the ants following the same routs on the loop we have used this test. The first curve shows ACO mode without loop removal and the second one denotes ACO mode with loop removal. It was observed that ACO performs better with loop removal strategy.

VIII. CONCLUSION

In this project, an exposition of the basic problem-solving paradigm of ACO was given. The differences between ACO and traditional routing algorithms were compared along the issues of routing information, routing overhead and adaptivity. Further an approach of using ACO for dynamic routing on any communication network is presented. The routing tables are generated on the basis of the pheromone count being updated continuously. The results thus obtained with ACO mode of routing are compared with the Non-ACO mode of routing in a simulation and the results were plotted in terms of number of intermediate hops versus the number of calls made. On comparison it was found that the ACO mode always results in reduction of the intermediate nodes while in completion of the calls. Thus ACO provides afield for future research in application of AI in routing.

IX. REFERENCES

[1] Marco Dorigo, Mauro Birattari, and Thomas Stutzle, "Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique", IRIDIA – Technical Report Series, 1-12, 2006

[2] S.Venkatesan and Dr.S.Srinivasa Rao Madane, International Journal of Innovation, Management and Technology, Vol. 1, 5, 2010

http://www.ijimt.org/papers/82-M477.pdf

[3] IEEE Mechatronics and Automation,2009.ICMA 2009 Zheping Yan ; Yanchao Zhang ; Xiaomin Fu ; Shuping Peng's research on Genetic Algorithm and Ant Colony Optimization from

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5246484&queryText%3Dant+colony+optimization, 2009

[4] Study on immunized Ant Colony Optimization by Wei Gao, Natural Computation 2007.ICNC 2007 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=4344780&queryText%3Dant+colony+optimization, 2007

[5] Lixin Tang,Jiyin Liu,multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex,European Journal of Operational Research, 2000.

[6] Nada M. A. Al Salami, "Ant Colony Optimization Algorithm", UbiCC Journal, Vol. 4, 3, 2009

http://www.ubicc.org/files/pdf/10_336.pdf

