

ANTNET: A ROBUST ROUTING ALGORITHM FOR DATA NETWORKS

SHARIQ HASEEB*, KHAIRUL AZAMI SIDEK*, AHMAD FARIS ISMAIL*, LAI W.K.+ & AW YIT MEI†

* Faculty of Engineering, International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia.

+ MIMOS BERHAD, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia
e-mail: faris@iiu.edu.my

Abstract: Successful implementation and operation of a network largely depends on the routing algorithm in use. To date, several routing algorithms are in use but the problem with these algorithms is that they are either not adaptive or not robust enough, thus limiting the proper use of bandwidth. AntNet is an innovative algorithm that may be used for data networks. It is a combination of both static and dynamic routing algorithms. In this algorithm, a group of mobile agents (compared to real ants) form paths between source and destination nodes. They explore the network continuously and exchange obtained information indirectly, in order to update the routing tables at different nodes. Our version of AntNet (hereinafter referred to as AntNet2.0) has been improved to overcome the problems with other algorithms. This paper compares the performance of AntNet2.0 against two other commercially popular algorithms, viz. link state routing algorithm and distant vector routing algorithm. The performance matrix used to compare the algorithms is based on average throughput, packet loss, packet drop and end-to-end delay. Convergence time for this algorithm on a nation-wide telecommunications network will also be discussed. Conclusions and areas of further work will also be presented in lucid manner, so that it may be transformed into real practice in the future.

Key Words: mobile agents, swarm intelligence, networks and constant bit rate

1. INTRODUCTION

Swarm intelligence is a novel solution to many constraint optimization problems such as the Traveling Sales Person Problem, Quadratic Assignment Problem, Job Scheduling Problem and the Network Routing Problem. Swarm intelligence is based on the behavior of insects that normally work in groups, such as ants and bees [1]. These insects cannot communicate with each other directly, so they have to use their surroundings as a means of interaction. They communicate by laying down a chemical called pheromone that can be sensed by other ants and thus attracting them towards it [2].

In Nature, a traveling ant lays some pheromone (in varying quantities) on the ground, thus marking the path by a trail of this substance. While an isolated ant moves essentially

at random, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The collective behavior that emerges is a form of *autocatalytic* behavior, when there is a higher number of ants following a trail, the more attractive that trail becomes to the newer ants. The process is, thus, characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that has previously chosen the same path.

This ability of ants to collectively seek a better path can clearly be reflected in many constraint optimization problems, such as the network routing problem. In this problem, the constraints are the links and their capacities. The most efficient route is chosen, in order to achieve the best performance. The performance of a data network reflects the speed, reliability and capacity of the network, as overloaded networks may easily be disrupted by attacks.

2. ANTS AND PATH SELECTION PROCESS

Small insects like ants have only local knowledge. When a group of such insects interact with each other, they can exhibit a higher-level behaviour. This behaviour is most often called emergent behaviour. A group of insects like ants do not have a central controlling authority that steers the behaviour. All ants behave according to a few simple rules and by interacting with their environment. The resulting behaviour of the group of ants can be very complex [3].

The important aspect behind emergent behaviour is called stigmergy. Stigmergy is a way for entities to communicate indirectly with each other through the environment. Many social insects including ants use stigmergy. One of the abilities of a group of ants is to find the shortest route from their nest to a food source. Ants only react to local stimuli from the environment and can change one or more of those local stimuli. This modification can be of two types. Firstly, physical, in which behaviour of one ant influences the others to follow, i.e. an ant seen carrying food by other members of its colony, will influence these members also to carry food, hence resulting in a collective work being done. Secondly, sign-based, in which one ant lays down a hormone trail as it moves resulting in the other ants to follow. The goal of sign-based stigmergy is to influence subsequent behaviour. Ants are very good in using this second method [3].

Ants can lay pheromone trails when going from the nest to the food source, from the food source to the nest or in both directions. When looking for food, ants follow the pheromone trails with *about* the same probability as the strength of the trail. They do not follow the trail with precisely the same probability, but with a certain amount of error also known as noise. The strength of the trail sensed by an ant depends on the original strength and the time elapsed since the trail was laid, since pheromones diffuse away over time. Multiple ants can travel at the same route, so the resulting trail can consist of multiple trails laid by different ants at different times. The pheromone trail sensed by ants is, therefore, one that has been laid down by many ants, i.e. a composite trail [2].

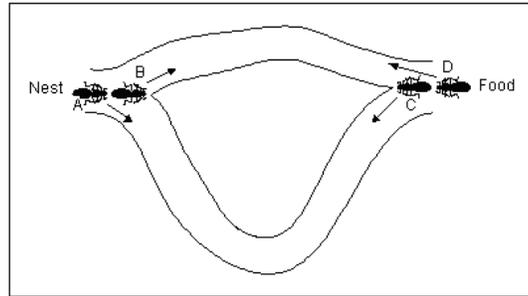


Fig. 1: Paths between nest and food.

The four ants in a situation depicted in Fig. 1 have a decision to make. Ants A and B come out from the nest to the food source and the ants C and D come out from the food source to the nest. The ants in this example lay trails that influence the ants' decisions in both directions. Each ant has a choice between two routes from the nest to the food or vice versa. When no pheromone trails have been laid, the ants arriving at one of the branches have an equal probability of 0.5 of choosing either the left or the right route.

Ant B that travels from the nest to the food takes the left route. The ant following ant B, ant A, already senses a faint pheromone trail from the first ant. However, the influence of this faint pheromone trail is still small, because the ants use a sufficient amount of noise to make them explore new routes. This means that the probability of ant A choosing either the left or the right route is still close to 0.5. In this example, ant A chooses the right route. Ants C and D that are travelling from the food to the nest take the same amount of steps. Ant C chooses the left route and ant D chooses the right route.

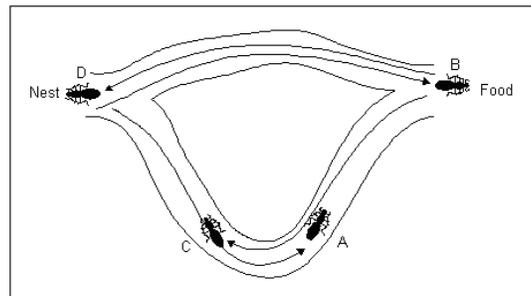


Fig. 2: The ants have laid their pheromone trails.

The two ants (B and D) that choose the shortest route have arrived at their destination as shown in Fig. 2. The other two ants, ant A and C, are still travelling along the long route and are expected to arrive at their destination in due time.

From Fig. 2, it is clear that the shorter path between the nest and the food has a higher concentration pheromone at this particular instant. So, if new ants want to select a path either from the nest to food or the food to nest, there is a higher probability that they will be attracted to the shorter path because a stronger pheromone trail increases the chance of an ant choosing a route belonging to that trail. If this happens, they will further concentrate

the trail on the shorter path. The ants will be attracted towards the shortest paths, because of the following three reasons:

- Shorter routes will be completed earlier than longer routes and thus attract other ants to their source nodes first.
- Shorter routes involve fewer branches, so the number of ants will be larger than on longer routes with more branches.
- Ants travelling shorter routes will be younger when they arrive. This causes the pheromone trails to be stronger, because less of the pheromone trail has diffused away.

When more and more ants start taking the shorter path, the concentration of pheromone on the shorter path becomes more concentrated. As a result, fewer and fewer ants take the longer path, until finally no more ants get attracted to the longer path. When this happens, the concentration of pheromone on the longer path tends to decrease, until finally, it completely disappears.

3. FROM NATURAL TO ARTIFICIAL SYSTEM

The behavior of ants represents a unique method of finding the shortest path between two locations. The setup of Fig. 2 provides a solution to network routing problem too, however this system had to be modeled differently in order to represent a problem solving technique.

M. Dorigo[3] proposed the following modeling of the natural system that allowed network routing problems to be solved:

- The ants were modeled as mobile agents or routing packets.
- Different locations on the ground were modeled as network nodes.
- Pheromones were modeled as probabilistic values.

4. THE ANTNET ROUTING ALGORITHM

There are two types of ants used in this algorithm, namely, *Forward ants* and *Backward ants* [5]. These ants can be compared to the ants leaving the nest for food and ants returning to nest from food, in the ant nest simulation explained earlier. The forward ants have the same priority as the normal data packets in a network. They accomplish their task by the help of a timer that measures the amount of time spent on a link. When forward ants reach their destination node, they become backward ants. These backward ants have a higher priority to travel the network. The reason for this is that they carry useful routing information that must be transferred to the routers immediately. Thus, backward ants update the routing table of a node.

It is vital for these two ants to interact with each other in order to share information. This information sharing can only be done in an indirect way. Live ants interact with each other by laying down pheromone trails on the ground. However, a network is not exactly like a path. It is a collection of nodes represented by routers and links are represented by wires as physical connections or wireless. The only place where digital pheromone can be laid down upon is the routers because the physical links have no memory of their own.

Figure 3 shows a typical content of a node in a network. It contains a routing table showing the digital pheromone (P11, P12, P13, ..., Pnn) concentration between different nodes and local traffic statistic table containing information about the interconnections of other links on the network.

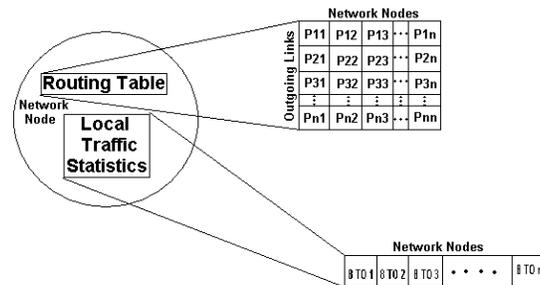


Fig. 3: Data Structure of a node in the network [4,5].

4.1. ROUTING TABLE CONSTRUCTION

The idea of emergent behaviour of natural ants can be used in telecommunication networks to build routing tables. Telecommunication networks cannot sense the pheromone trails, but they can use probabilities. The routing tables will be called probability tables from this point onward. Each node in the network has a probability table for every possible final destination. The tables have entries for each next (neighbour) node. The probabilities influence the agents' selection of the next node in their journey to the destination node. The probability of the agents choosing a certain next node is the same as the probability in the probability table.

The probability tables only contain local information and no global information on the best routes. Each time an agent visits a node, the next step in the route of the agent is determined. This process is repeated until the agent reaches its destination. Thus, the entire route from a source node to a destination node is *not* determined beforehand. The probability tables also contain information if a node is reachable or not. This information is contained in the last column of each table (Fig. 4). If an agent detects that a node is unreachable, it will mark the entry for this destination node unreachable in the probability table of the agent's source node. Agents continue to be launched while old agents detect that a node is reachable again. The reachable node will then be marked reachable in the probability table of the agent's source node.

Table 1: An example of a probability table.

Node 1	2	4	R
2	0.95	0.05	Y
3	0.67	0.33	Y
4	0.05	0.95	Y
5	0.83	0.17	Y
6	0.73	0.27	N

First column of Table 1 shows all the other nodes to which node 1 can send packets. The first row of the table shows the nodes to which node 1 is connected and 'R' represents the reachable status between node 1 and the destination nodes of column one, where 'Y' stands for reachable and 'N' stands for not reachable, in column four. The probability values show the attractiveness of selecting the neighboring nodes. Even though node six is not reachable, we noticed some probability values, which were assigned to the neighboring links before the path to node six was broken.

4.2. ROLE OF ANTNET AS A LOAD BALANCER

In conventional algorithms, once a best route is selected, all the packets are routed along that route. This causes the link to become overloaded and thus decreasing its performance dramatically. In some cases, it might even cause the link to fail altogether resulting in 100% packet drop on that network. However, AntNet has a mechanism for load balancing, where after selecting the best link, it sends its packets on it. It is different from the conventional algorithm in the sense that, as soon as a packet is released on the link, the sending node reduces the probability value in the routing table for the optimal link and increases the probability values of the other non-optimal links. If this process keeps on going, there will be a situation when the less desirable link will have a higher probability and thus making it the optimal route. When this happens, the next packet will now be routed on the new optimal path and this prevents the links from overloading.

4.3. PERFORMANCE METRIC

We assume that all nodes can reach each other by some route through the network. There are two main performance criteria for network routing algorithms. The first one is effectiveness and the second one is efficiency. Effectiveness means how good the algorithm is capable of doing its job, and efficiency means how well the resources are used. The parameters we used to test the algorithm can be classified as either effectiveness or efficiency metric as seen below.

4.3.1. Effectiveness:

- Average Throughput: a throughput value is the number of bytes sent in the network per second. This calculated value subtracts the dropped and lost bytes from the received bytes.

- Packet Drop: this value represents the number of packets dropped in the network.
- Packet Loss: this value represents the number of packets lost on the network during transmission.

4.3.2. Efficiency:

- Delay: delay value is calculated by subtracting the time at which a packet left the source node from the time at which it reaches the destination node.

4.4 EXPERIMENTAL SETTINGS

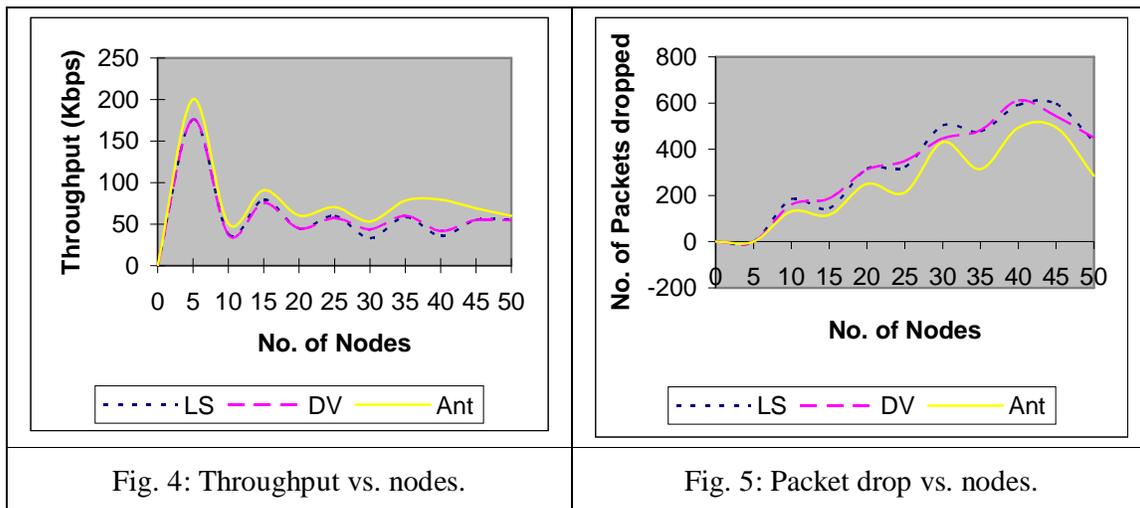
To test the AntNet 2.0 algorithm, several test networks were made. In these test networks, number of nodes was increased from 5 until 50 with increments of 5 nodes at a time. The link capacity between nodes was set to 2 MB/sec and the default delay of each link was 10ms.

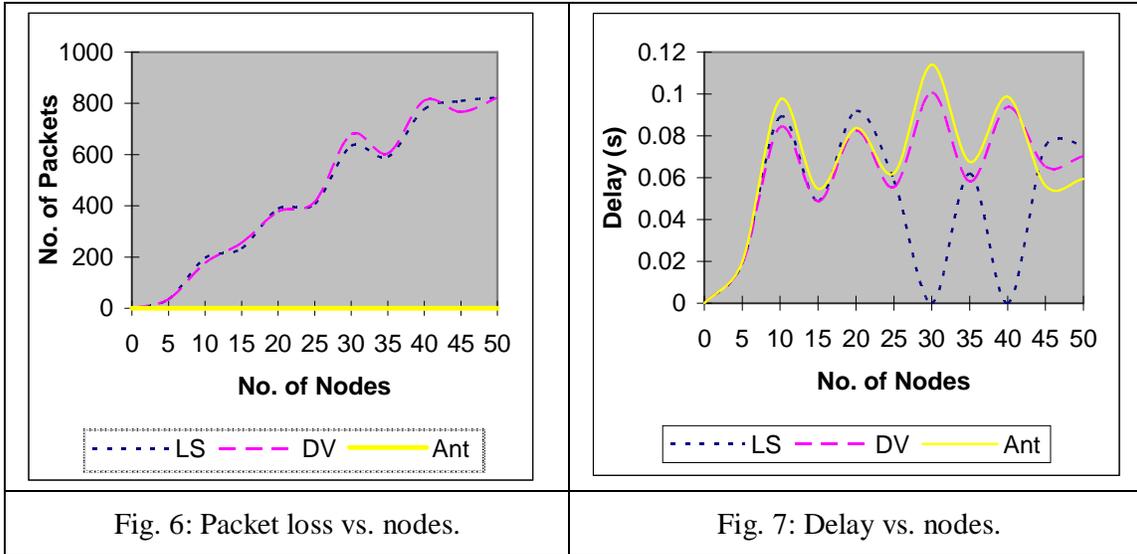
Each generated network was tested with two types of traffic, namely Transfer Control Protocol (TCP) and Constant Bit Rate (CBR). Each simulation was run for 5 simulation seconds. In order to make comparisons, these settings were also used to simulate the other algorithms like Distant Vector (DV) and Link State (LS).

This setup is used because it provides variance to the network architecture and results in a more accurate result. As the number of nodes increases, the number of links connecting the nodes have to increase resulting to more paths available. An increase in the number of nodes also results in higher traffic, thus providing a solid platform for comparing the algorithms according to the following performance matrix: throughput, packet drops, packet loss and delay.

5. RESULTS

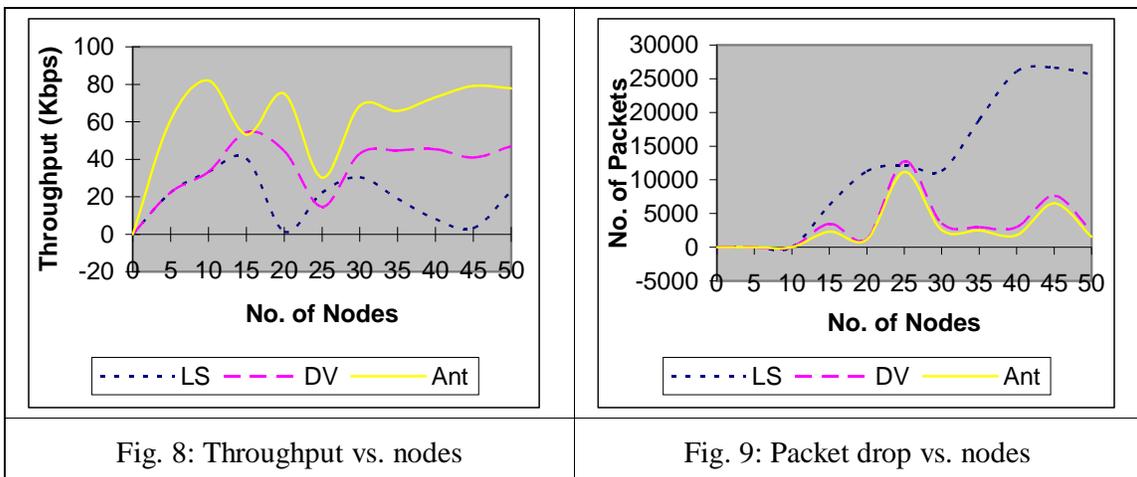
Results for TCP traffic sources are shown below:

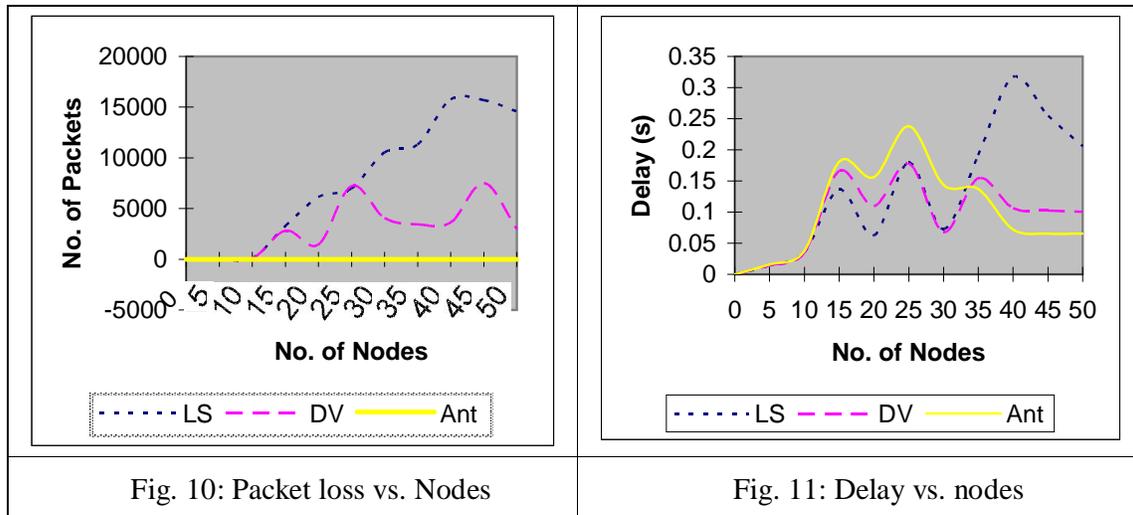




The above results were obtained by simulating the network topologies with a TCP source. TCP has variable bit rate and built-in flow control, as the new packets are only sent if acknowledgement packets for the sent packets have reached the source node. Testing networks with TCP sources is justified because real world networks are normally based on variable bit rate conditions. From the graph shown in Fig. 4, one can infer that the throughput produced by AntNet 2.0 is better than that produced by LS and DV. Figure 5 shows that AntNet 2.0 has a lesser packet drop, as compared to LS and DV. It is visible from Fig. 6 that AntNet 2.0 has no packet loss while the other algorithms have a significant degree of packet loss. In a constraint optimization problem, one has to compromise on something in order to gain something. This is the case illustrated in Fig. 7, where AntNet produce more delay than other algorithms.

Results for CBR traffic sources are shown below:





The results shown in Figs. 10 and 11 were obtained by simulating the network topologies with a CBR source. CBR has constant bit rate and no flow control, as the new packets are sent and no acknowledgement packets are expected. Testing networks with CBR sources is equivalent to doing an acid test of the network. It is justified because it shows the analyst how much data can be transferred, before the networks fails or is too overloaded to be used. From the graph shown in Fig. 8, one can infer that the throughput produced by AntNet 2.0 is much better than that produced by LS and DV. Figure 9 shows that AntNet 2.0 has much lesser packet drop, as compared to LS and DV. It is visible from Fig. 10 that AntNet 2.0 has no packet loss while the other algorithms have a high degree of packet loss. The result obtain in Fig. 11 is extremely unique because AntNet 2.0 should produce higher delay but for higher number of nodes, its delay is less than that of LS and DV. A simple explanation for this is that when we had higher number of sending nodes, the network became overloaded and since other two algorithms have no load balancing technique, they became too overloaded, which caused excessive delays.

TIME TO CONVERGE

The previous section showed that AntNet2.0 algorithm is more robust, scalable and reliable than the other two algorithms. However, another important factor that hinders the use any such algorithm in commercial networks is the convergence time factor. Convergence time refers to the time that an algorithm takes so that all the nodes in the network have complete routing table. To test this phenomenon, we ran our algorithm on the simulated version of the Jaring backbone network shown in Fig. 12. The data to construct the replica of the network was provided to us by the Jaring representatives. Figure 13 only gives a rough idea of the topology, as more accurate information is classified and no third party is allowed to have access to it.



Fig 12: Jaring backbone network.

In our experiment, the network of Fig. 12 was routed repeatedly without changing any of the original conditions and the time to reach a convergence state was noted each time. When this network was routed for the first time, it took a lot of time to reach convergence, but it can be noticed from Fig. 13 (values taken from Table 2) that, as the network got routed more number of times, time to converge reduced exponentially. This was observed because the mobile agents produced in the first route, pass their information to the second batch of mobile agents generated in the second route. Thus the newer agents have to traverse fewer areas of the network in order to construct the routing table.

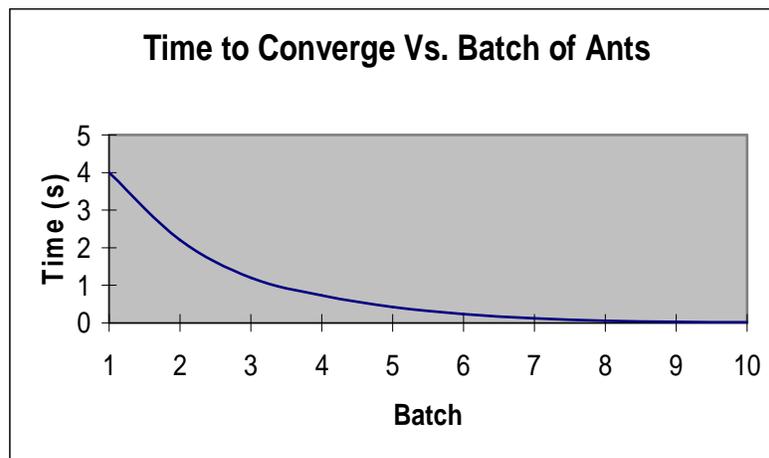


Fig 13: Convergence times.

Table 2: Convergence times.

Batch	Time (s)
1	4
2	2.2
3	1.2
4	0.73
5	0.42
6	0.2351
7	0.1185
8	0.05935
9	0.029685
10	0.014844

6. CONCLUSIONS AND SUGGESTIONS

In conclusion, it is to be suggested that the AntNet 2.0 algorithm performs better than the other algorithms in use today. However, it is to be noted that AntNet 2.0 produces about 20 to 30 percent more delay than the other algorithms. This might deter the network engineers to implement the algorithm as it is today. Future works in the area of reducing the delay may result in this algorithm being adopted widely by the private networks.

REFERENCES

- [1] M. Dorigo, V. Maniezzo, A. Coloni. 'The Ant System: Optimization by a colony of cooperating agents'. Pp 1-13. vol. 26-Part B. IEEE Transactions on systems, Man, and Cybernetics. 1996.
- [2] E. Bonabeau, M. Dorigo, and G. Théraulaz, *Swarm intelligence: from natural to artificial systems*, Oxford University Press, 1999.
- [3] M. Dorigo, Luca Maria Gambardella. *Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem*; IEEE Transactions of Evolutionary Computation, 1:53-66(1), 1997.
- [4] *Networks and routing - A survey*; Aronson. Technical Report 94-112, Faculty of Mathematics and Computer Science, Delft University of Technology, 1994.
- [5] G. Di Caro, M. Dorigo. "AntNet: Distributed Stigmergetic Control for Communication Networks". *Artificial Intelligence*, 9, 317-365, 1998.
- [6] Ruud Schoonderwoerd. *Collective Intelligence for Network Control*; Master thesis, Faculty of Mathematics and Computer Science, Delft University of Technology, 1996.
- [7] Ruud Schoonderwoerd, Owen Holland, Janet Bruten, Leon Rothkrantz. *Load balancing in telecommunication networks*; *Adaptive Behaviour*, 5(2), 1997.

BIOGRAPHIES

Shariq Haseeb obtained his Bachelor of Engineering Degree from International Islamic University, Malaysia. He is currently pursuing MSc (Computer and Information Engineering) in International Islamic University, Malaysia. His research interests are in swarm intelligence, mobileIPV6, active networks and other computer network related areas.

Khairul Azami Sidek was born in Melaka, Malaysia in 1980. He received his B.Sc. degree in Computer and Information Engineering from the International Islamic University Malaysia in 2003. He is currently an Assistant Lecturer at the Department of Electrical and Computer Engineering at the International Islamic University Malaysia. His primary research interests are networking, software development, database management system and internet applications.

Ahmad Faris Ismail currently is a professor and the dean of faculty of engineering at International Islamic University Malaysia (IIUM). He obtained his Bachelor of Science in Chemical Engineering from University of Houston in 1988 before completing his Ph.D. from Rice University, U.S.A. in 1993. His research areas include discrete modeling, energy systems, combustion, computational fluid dynamics, engineering mechanics, computer vision, and image processing. He has published more than 70 technical papers in these areas in international journals and conference proceedings.