An Improved Subset AntNet Algorithm for Numerical Optimization of the Shortest Path Problem

Raghad Zuhair Yousif, Ahmed Tariq Sadiq

Abstract— Routing algorithm is the key element in networks performance and reliability, therefore, there is a need for an algorithm to manage traffic flows and deliver packets from the source to the destination in a realistic time. A modification algorithm has been presented called Subset Nodes Based AntNet. In the Subset Nodes Based AntNet, a routing tables maintained at each node is used to get a subset table which helps in reducing number of choices available to each Ant to select the next node and reach the destination. The parameters used to measure the performance of the proposed routing algorithms are: cost, number of iterations and number of Ants. The experiment results show that the proposed Subset Nodes Based AntNet is best in minimizing the cost, number of iterations and number of Ants number in comparison to the original AntNet.

Index Terms— Bioinformatics , AntNet, Computer Networks , Routing problem,Subset Antnet,Ant Colony,Visual c++ .

1 INTRODUCTION

An ideal routing algorithm should be node and link independent, and able to deliver packets to their destination with the minimum amount of delay, regardless of the network size and the traffic load. The only way to achieve this goal is the employing of an intelligent and distributed routing algorithm [1]. With the growing importance of telecomm unication and the Internet, more complex networked systems are being designed and developed. The challenges of dealing with the vast complexity of networking problems such as load balancing, routing and congestion control accentuate the need for more sophisticated (and perhaps more intelligent) techniques to solve these problems. Drawing upon some of the computing techniques inspired by social insects such as ants [1],which was derived from Swarm Intelligence (SI) which is the local interaction of many simple agents to achieve a global goal[2] , several mobile agent-based paradigms were designed to solve control and routing problems in telecommunication and networking. Although 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) [1].Ants use pheromone for communication in a number of contexts, including alerting other individuals to a threat, recognizing individuals from the same colon, marking trails to food sources and recruiting other ants to collect food from those sources [3].

Routing is fundamental in communication network control. In data networks it means the action of addressing data traffic between pairs of source-destination nodes. The routing task is performed by routers, which update their routing tables by means of an algorithm specially designed for this purpose. The first routing algorithms addressed data in a network minimizing any costs function, like physical distance, link delay, etc. [4][5]. As a result, traditional routing methods do not have enough flexibility to satisfy new routing demands.

New network services and the impressive increase in the amount of users force network administrators to improve throughput in order to satisfy the immense amount of simultaneously requested services. This situation has impelled the study and development of other routing methods, to satisfy these new demands [6]. Actually, Algorithms that take inspiration from real ants' behavior in finding shortest paths using as information only the trail of a chemical substance (called pheromone) deposited by other ants, have recently been successfully applied to several discrete optimization problems. Using AntNet, the algorithms we propose in this paper, based on generating a set of concurrent distributed agents which are collectively solve the adaptive routing problem. AntNet shows the best performance and the most stable behavior for all the considered situations. Foraging is a very important behavior of the ant colony. This behavior is closely connected to the fact that ants are able to find the shortest path between their nest and the food source. While searching for food, a forager deposits pheromone on its path. In this way the ant is marking its path. When it finds food, it can go back to its nest using the same path. Other ants can also smell this pheromone and follow the trail that would lead them to the food source. Through the experiments it's proved [6][7] that the path with a higher concentration of pheromone is more attractive to the ants than the path with a poorer concentration of pheromone. When a path is not used for a while, the pheromone concentration can decrease due to evaporation. This path is becoming less attractive to an ant to use. Otherwise, a path that is used more frequently will always get more pheromone. Having a higher concentration of pheromone, this path will be more attractive to the ants telling them that the path is connecting the nest with a recently found food source. The principle of finding the shortest path can be explained with the following procedure [7]. We will consider a simple model with two possible paths between the nest and

- **Raghad Zuahir Yousif** ,Department of Applied Physics-Communication, Salahaddin University ,College of Science , Erbil, Iraq, (e-mail: DrRaghad.Zuhair@gmail.com).
- **Ahmed Tariq Sadiq,** Department of Computer Science, University of Technology, Baghdad, Iraq, (e-mail: AhmedT@yahoo.com).

the food source1 with one T-junction A near the nest and another one D close to the food source (figure 1). The lower path is shorter than the upper path. We will consider a situation with four ants, two coming from the nest and two returning to the nest.

If there is no pheromone presented on the paths, an ant coming to a junction point will choose randomly, with the same probability, the upper or the lower path. We suppose that on each side one ant will choose the lower path and the other ant the upper path. As they are walking, they are leaving behind a pheromone trail (given with a line that leads to an ant in figure 1 part b). The ants on the lower path will complete their journey faster than the ants on the upper path. On this path, the concentration of pheromone is higher than on the upper path (figure 1 part b). If, in the situation given in figure 1, part b, another ant comes to a T-junction (A or D) then it's more probable that it is going to choose the lower path.

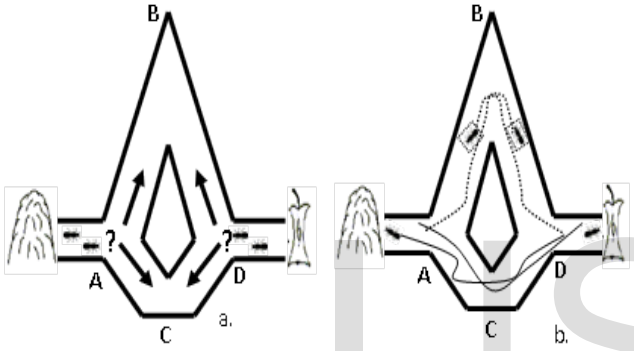

Fig.1 Schematic principle, of choosing the shortest path. First ants are choosing randomly (a); the shorter path will be accomplished faster (b).

Ths ant is not choosing randomly, its decision is biased by the pheromone concentration on the path. Thus, this ant will use the lower (shortest) path, and in this way the pheromone concentration on this path will grow further. If now more ants come from the nest or go back to the nest, it is to expect that their decision would be influenced by the pheromone trail on the paths, and that the majority will choose the lower path. The final result is that soon enough, all ants will select the shorter path. From this example we can conclude that, the shorter path will be completed faster than the longer path, and the pheromone concentration is growing faster on the shorter path and it will attract other ants earlier - this gives a link between the shortest path and the pheromone concentration;

2 ANTNET ALGORITHM IN SOLVING ROUTING PROBLEM MISSION

AntNet is an ACO algorithm for distributed and traffic-adaptive multipath routing in wired best effort IP networks. AntNet behaviour is based on the use of mobile agents, the ACO's ants that realize a pheromone-driven Monte Carlo sampling and updating of the paths connecting sources and destination nodes.

Informally, the AntNet algorithm and its main characteristics can be summarized as follows [5][8]:

• At regular intervals, and concurrently with the data traffic, from each network node mobile agents are asynchronously launched towards randomly selected destination nodes.

• Agents act concurrently and independently, and communicate in an indirect way, through the information they read and write locally to the nodes.

• Each agent searches for a minimum cost path joining its source and destination nodes.

• Each agent moves step-by-step towards its destination node. At each intermediate node a greedy stochastic policy is applied to choose the next node to move to. The policy makes use of:

(i) Local agent-generated and maintained information,
(ii) Local problem-dependent heuristic information, and
(iii) Agent-private information.

• While moving, the agents collect information about the time length, the congestion status and the node identifiers of the followed path.

• Once they have arrived at the destination, the agents go back to their source nodes by moving along the same path as before but in the opposite direction.

• During this backward travel, local models of the network status and the local routing table of each visited node are modified by the agents as a function of the path they followed and of its goodness. Once they have returned to their source node, the agents die.

2.1 Data structure of a node

The nodes in the network are functioning also as routers. To accomplish this task, they are using ants to gather information about the traffic load in the network. The traffic load depends on the amount of packets in the network (congestion level of the network). The congestion level is proportional to the delay that a data packet experiences during the trip from its source to its destination. If the congestion level is high the data packets will need more time to get to their destination. To collect the information about the traffic load on their path towards destination, the ants are calculating the time that a data packet would experience using the same path. This information is used to update the two data structures in a node. These two data structures are: the routing table and the local traffic statistics (figure 2). The ants are able to read and write in these two structures2, while the data packets are only reading information from the routing table to get to their destination. Routing table is a local data-base that helps router to decide where to forward data packets. It contains the information which specifies the next (neighbour) node that should be taken by a data packet to get to any possible destination in the network. Each routing table is organized as a set of all the possible destinations (all the nodes in the network), the probabilities to reach these destinations through each of the neighbours of the node (next hops). A routing table in a node i is given as $P_i = \{p_{jd}\}$ with p_{jd} is the probability value that expresses the Goodness of choosing node j as its next node from the current node i if the packet has to go to the des

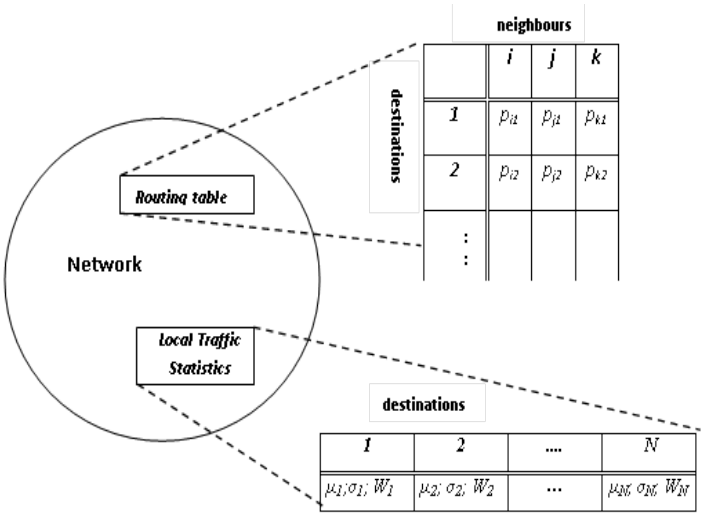

Fig.2 the data structures in a node: routing table and local traffic statistics. The presented node has i, j and k as its neighbours, and the destinations are all nodes in the network. The network has N nodes

tination node d. For any connected node i it holds that $\sum j P_{jd}=1$; where j is a neighbour of node i and d is an arbitrary destination from all nodes in the network From the ant colony point of view [1] , [2] , [5] , [8]. The probabilities in the routing tables can be seen as amount of pheromone. The probabilities are the product of continual exploration process of the ants. They are updated by the ants that previously used a path that is leading to the same destination.

Local traffic statistics is a second data-structure that each node has. The main task of this structure is to follow the traffic fluctuations in the network.It is given by an array Mi (μ_d, σ_{d2}, W_d) that represents a sample means μd and variance σ_{d2} computed over the packet's delay from node i to all the nodes d in the network, and value Wd where the packet's best trip time towards destination d is stored. The array Mi contains statistics about the traffic from node i towards each possible destination d. The mean μ_d and variance σ_{d2} are giving an expected trip time and its stability for each possible destination in the network. W_d is obtained by applying a moving observation window of size x to store the agents' trip time to destination d. This window is used to compute the best agents' trip time towards destination d W_{best} as observed in the last x samples. Wbest represents a short term memory and it should follow the fluctuations of the traffic in the network. Obviously, this cannot be the best trip time, but only a moving (temporary) lower bound of the time needed to travel from the current node to some destination node d. The values from the local traffic statistics are used for estimation of the quality of the ant's path. These values are used to update the routing tables.

2.2 Updating of the data structures in a node

Adscription of how the local data traffic and a routing table are updated with respect to a destination node d is presented in this section [2],[5]. To update these two data structures in node, virtual delay (packets trip time computed by the forward ant) has been used. This delay is used as a measure of the quality of the path that the ant has used. The virtual delay is proportional to the length of the made path from two points of view: The traffic congestion point – the virtual delay depends on the number of packets in the nodes on the used path. If there are more packets on the ant's path towards its destination, the virtual delay will be longer. Otherwise, for the fewer packets, the delay will be shorter. The physical point of view – the delay depends on the number of hops, transmission capacity of the links. If the ant makes (physically) longer path the delay will be longer. The update procedure is as follows. In order to update the local traffic statistics in a node i, a backward ant is using the information stored in its memory. The virtual delay needed to get to the destination d, starting from the current node i is used to update the mean μd, the variance σ_{d2} and the best value over the observation window W_d. To update the mean μd and variance σ_{d2} the following formulas are used [2][5][9].

$$\mu'_d = \mu_d + \eta (T_{i\to d} - \mu_d) \quad (1)$$

$$\sigma'_{d2} = \sigma_{d2} + \eta ((T_{i\to d} - \mu_d)^2 - \sigma_{d2}) \quad (2)$$

where: μ'_d and σ'_{d2} are the new (updated) values of the mean and the variances of the virtual delay, respectively; μ_d and σ_{d2} are the old values of the mean and the variances, respectively; $T_{i\to d}$ is the virtual delay of the observed agent from node i to destination d; η is the factor that weights the number of recent samples that will affect the average. It is experimentally found as $\eta \approx 5*(1/$ (number of effective observations)). For example, for the latest 50 observations it is 0.1. The values computed here will be used to update the routing table in the node. The routing table changes by re-evaluation of the probability of choosing the neighbour j when the destination of the packet is d (given as P_{jd}), and normalizing other probabilities that are leading to the node d. The reinforcement of the probabilities is proportional to the quality ('goodness') of the build path, where the quality of a path is a function of the virtual delay $T_{j\to d}$ and the local stochastic model of traffic in the node. The probability P'_{jd} is the (new) reinforced probability value, calculated with the following rule [5][9].

$$P'_{jd} = P_{jd} + r(1-P_{jd}) \quad (3)$$

Where Pjd is the value of the probability before reinforcement and r is the reinforcement factor. From the equation 3, we see that the value of the probability is increasing proportionally to the received reinforcement. This means that, if we have the same (constant) reinforcement factor r, the small probability values are increasing more than the high probability values. In this way the probabilities will never get too high values (too small values) that would attract ants to use always the same path, but it is also giving them a chance to discover new paths. The reinforcement applied to the routing table has been derived from experimental results which show that r=0.5 give best results, and are intended to limit the effect of the traffic fluctuations in the network at a given time.

3 PROPOSED SUBSET NODES BASED ANTNET

This proposal starts by generating a new Ant which inherited the memory of the original AntNet after an ant completes its tour to find the destination node and complete its routing table. First of all a selection of a source node has been made then the routing table of this node is constructed, and the simulation of proposed algorithm is started aimed to derive another routing table which is subset from the previous (initial) presented table. The column wise values in the table are picked up and a sorting algorithm is executed on these values to sort them ranging from higher to lower, the output of sorting is stored in a temporary array. The difference d, amongst the adjacent values are calculated and is compared with a threshold value p_m= 0.1. If the difference d between first two adjacent values in the certain column is less than pm then the values participating in this difference calculation are stored in the new routing table (selected), and comparison amongst the adjacent values is continued (the difference calculation is continued) until it becomes greater than pm. Otherwise at the very first occurrence of difference greater than pm, the comparison is stopped and the corresponding values in the array are selected. Finally the new routing table will contains all interfaces pass the proposed threshold test. The new (restricted) routing tables which will have the same structure as the original one, but obviously, is this table number of rows have been reduced due to the cancellation of some interfaces done by the proposed algorithm named (Subset Nodes Based AntNet Algorithm) Figure 3 shows general steps of the Subset Nodes Based AntNet:.

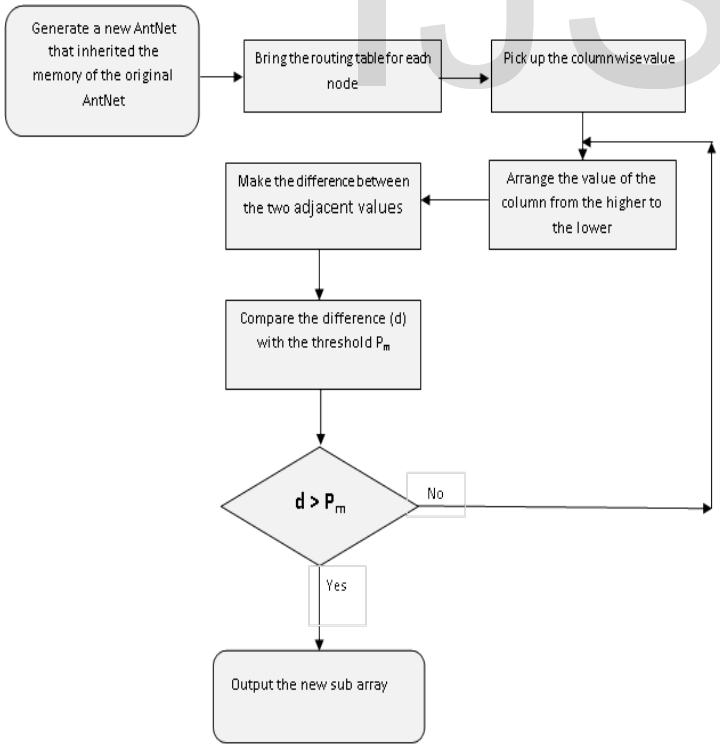

Fig.3 General Steps of Subset Nodes Based AntNet Algorithm.

3.1 Forward-Ant Algorithm:

The pseudo –code for Forward-Ant is illustrated below:

Algorithm 1: AntNet- Forward-Ant

Input (source node, destination node)
Output (become a backward-ant)
Begin
Initialize data;
K ← source node; /* k= current node */
For i = 1 to m do; /* m = number of ant */
For j = 1 to c do; /* c = number of nodes */
Setting all ants memory to false; /* memory ant is empty */
Endfor, Endfor;
For i = 1 to m do;
Assign ants to a random initial city; **Endfor**
V ← K; /* V = list of visited node */
While (K ≠ destination node);
n ← select next node; /* n = number of neighbors*/
K ← n;
If (k ∈ V) do /* check if ant is in a loop and remove it */
Cycle ← get cycle length (k, V);
Forward-ant ← Forward-ant – cycle length;
Else
Forward-ant ← Forward-ant +1;
V ← K;
EndIf
EndWhile
Generate a Backward-ant (V, R); /* R = routing table*/
End Procedure

3.2 Backword- Ant Algorithms:

The pseudo –code for backword-Ant is illustrated below:

Algorithm 2: AntNet- Backward-ant (V, R)

Input (V, R)
Output (updates R, update traffic model M_i, update pheromone matrix T_i)
Begin
K ← destination node;
Backward-ant ←Forward-ant;
While (K ≠ source node);
Backward-ant ← backward-ant **-1**;
n ← v[backward-ant];
go to next node;
K ← n;
For i = backward ant +1 **do** /* update all paths*/
Update traffic model M_i;
$\mu_{id} \leftarrow \mu_{id} + \eta\ (o_{i \to d} - \mu_{id})$;
$\sigma^2_{id} \leftarrow \sigma^2_{id} + \eta(\ (o_{i \to d} - \mu_{id})^2 - \sigma^2_{id}\ )$;
get reinforcement *(r);*
update pheromone table;
$P'id \leftarrow Pid + r.(1\text{-}Pid)$
update routing table;
EndFor
EndWhile
EndProcedure

3.3 AntNet Based Subset Nodes:

The pseudo –code for AntNet Based Subset Nodes t is illustrated below:

Algorithm 3: AntNet Based Subset Nodes

```
/*inherits the routing table from original AntNet
Input (updated routing table, pm)  /* pm = threshold */
Output (sub array contains all the optimal paths)
Begin
        For each node in the updated routing table   do
          Pick up the column from the nodes routing table
          Begin
                    For (i=0 OR i ≤(length of array -1) do;
                    Max = I;    /*max is a parameter to arrange */
                    Max =array[i];
                    For (j =i+1 and j≤ (length) do ;
              If (max≤ array[j]) ;
               Max=j;
               Max= array[j];
               End if
          End for
   Temp = array[i];
          Array[i] = array [maxat];    /swap/
          Array [maxat] = temp;
          Endfor
          d← the difference among the adjacent value;
           pm ← threshold value;
Calculate the difference among the adjacent value then compare it to the threshold value;
        If (d ≤ pm   ) do;
            Continue calculating (d) and storing their corresponding interfaces until   d ≥ pm;
            Else
When d ≥ pm;
            Stop comparison;
            Select the value and store its interface in the new array;
       End if
End Algorithm
```

4 Test Network

In Figure (4) below a simple network is presented to illustrate how the proposed routing algorithm works.

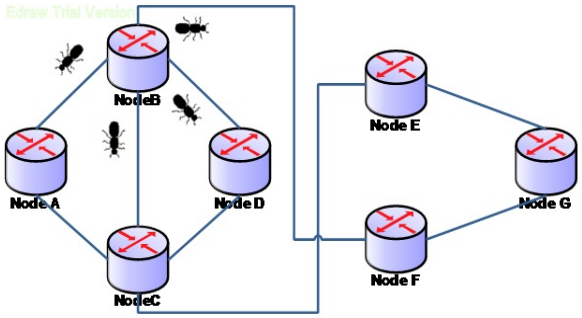

Fig .4 Test Network diagram

Consider a node B in the presented test network which has 4 neighbours. The routing table contains the probabilities list for node B next hope to reach every destination k. The probabilities are listed in the Table 1.

TABLE 1- ANTNET ROUTING TABLE AT NODE B

Outgoing Links	All Network Nodes (Possible Destinations)					
	PAE = 0.25	PAG= 0.39	PAF=0.8	PAA=0.34	PAD= 0.56	PAC=.47
	PCE = .45	PCG=.32	PCF =0	PCA=.34	PCD=.23	PCC=.43
	PDE = .17	PDG=.11	PDF=.1	PDA=.20	PDD=.01	PDC=.03
	PFE = .13	PFG=.28	PFF=.1	PFA=.12	PFD=.20	PFC=.07

For each column a calculation for generating sub array is repeated. Thus let we start with the 1st Column. The interfaces probabilities of the 1st column are picked up and a sorting algorithm is executed (ranging from higher to lower) and stored in the a temporary array say Ai, i.e. Ai ={0.45,0.25, 0.17,0.13}. The difference d =0.45-0.25=0.2.then is calculated and a comparison is made if (d > p_m) hence (0.2>0.1) , is True, and {0.45} is selected ,i.e. the interface corresponding to 0.45 (next hope is E)is stored in a new proposed routing table. The values of 2nd column in the table are picked up and a sorting algorithm is executed again. The sorted values ranging from higher to lower are stored in a temporary array. Ai, i.e. Ai ={0.39,0.32,0.28,0.11},the Difference is calculated as d =0.39-0.32=0.07,then a comparison is made if(d > p_m) in this case (0.07<0.1) is true , hence {0.39,0.32} are selected and the process is continued until (d > p_m) becomes true . Difference is calculated now as d=0.32-0.28=0.04, now (0.04<0.1) is true, hence {0.39, 0.32, 0.28} are selected. Then because (d > p_m) is false we have to calculate d=0.28-0.11=0.17, now(d > p_m) is true. Thus, the process of differences construction is discontinued, and the interfaces corresponding to {0.39, 0.32, and 0.28} are stored in a new proposed routing table. Consequently, we repeat the same calculations for the 3rd to the 6th columns. The results are summarized in Table 2 below:

TABLE 2- THE PROPOSED ROUTING TABLE AT NODE B

Destination	(Next-Hop, Probability)
E	(A, 0.45)
G	(A, 0.39), (C, 0.32), (F, 0.28)
F	(A,0.8)
A	(A,0.34),(C,0.34)
D	(A,0.56)
C	(A,0.47),(C,0.43)

It's clear from the table 2 more than one optimal paths if exist, are identified.

5 PROPOSED PACKAGE

Microsoft Visual C++ (often abbreviated as MSVC or VC++) is a commercial integrated development environment (IDE),it's a member of Microsoft Visual Studio 2008 family which has been used to build the simulation package .The proposed package is capable on simulating the original Ant-Net and the proposed modification for it(First update) in term of a certain parameter like minimum cost, best iteration, and best ant number.

5.1 The Simulation of Original AntNet

The first step in simulating original AntNet execution is started by selecting Original method check box: when this selection is made for a certain test network, the source and destination nodes must be assigned in their text boxes .A certain parameters must be entered numerically like No. of nodes, No. of Ants and the No. of iteration to construct optimum path between source and destination. Figure (5) illustrates test network with the assumption that: 15 nodes network, 8Ants, 9 Iterations, the source node =1 and destination node = 7

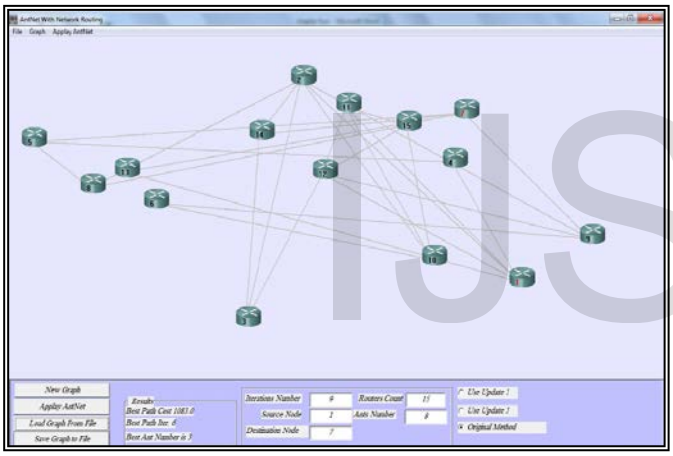

Fig .5 Execution of Original AntNet

The next step is illustrated in figure (6) involves the simulation of the original AntNet algorithm by clicking the Apply AntNet bottom depicted in to the previous figure:

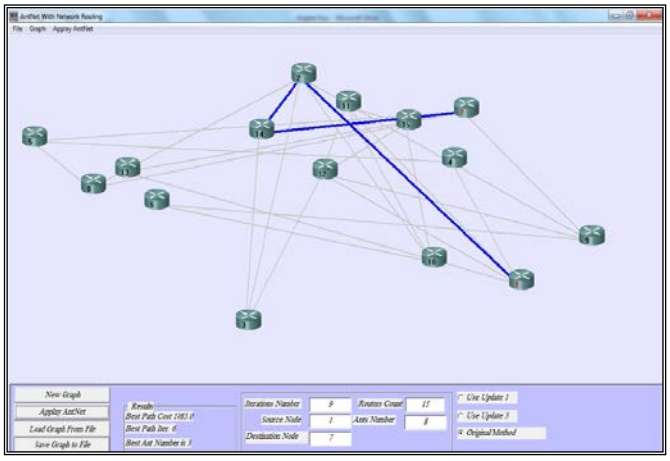

Fig .6 Applying the AntNet algorithm

As it illustrated in figure (6) at the end of simulation the proposed package establish blue colored lines between source and destination with the calculations of the minimum path cost, the best iteration numbers, and best ant number.

5.2 Simulation of Proposed Subset AntNet

The same previous procedure is repeated with unique difference which is instead of activating Original method check box the First update check box had been activated, which leads to simulate the Subset Nodes Based AntNet algorithm, with same previous parameters setting(in the case of original AntNet simulation depicted in pervious section) this is depicted in Figure (7)

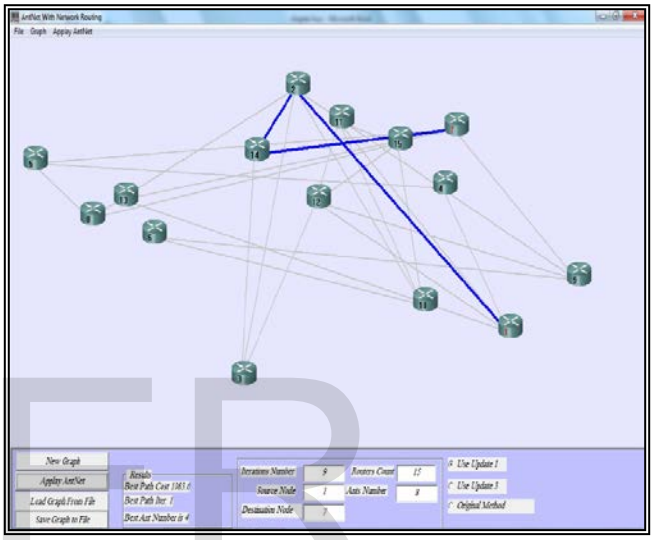

Fig.7 Execution of the proposed algorithm

As illustrated in Figure (7) the ant chooses the same path but the best path iteration and the best ant number are minimized.

6 EXPERIMENTAL RESULTS DISCUSSION

Figure (8) presents curves of comparison between original Ant Net simulation of finding optimum path between source and destination nodes when the number of iteration is half number of nodes. It is clear that increasing the number of nodes leads to increasing the number of iteration in the original AntNet, whereas in the case of the Subset Nodes Based AntNet, the number of iteration still constant with increasing the number of nodes because in this case a sub array has been constructed. Hence with less number of iterations needed to reach destination intended destination. Also from Figure (8) it's clear that after the number of nodes exceeds (20), the best iteration increased to (2) which mean that we need more iteration to find the destination. Hence there is a large gap between the number of iterations in the proposed Subset Ant-Net and the original AntNet.

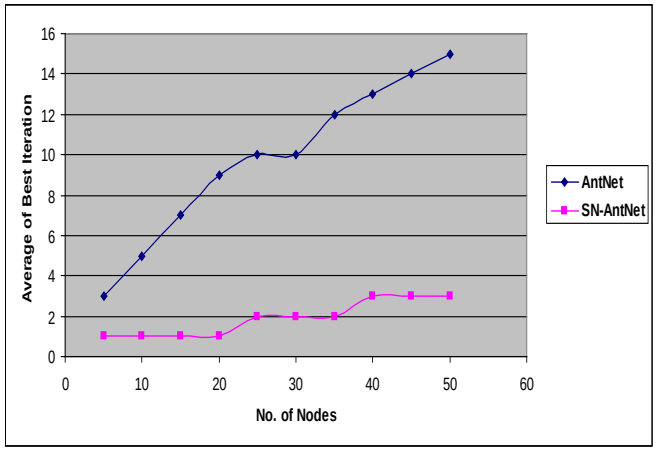

Fig. 9: Chart of Best Iteration Where Number of Nodes of Maximum Iteration is Equal to Three Quarter of The Number of Nodes.

Figure (10) illustrates the curves of comparison between the original and proposed Subset AntNet by investigating the relation between number of nodes and the best number of Ants when the number of Ant is half the number of nodes. Thus when the number of nodes increased in the proposed Subset AntNet the number of Ant is increased also. The proposed algorithm outperforms the original AntNet in decreasing the number of Ants needed especially when the number of nodes exceeds 30 nodes.

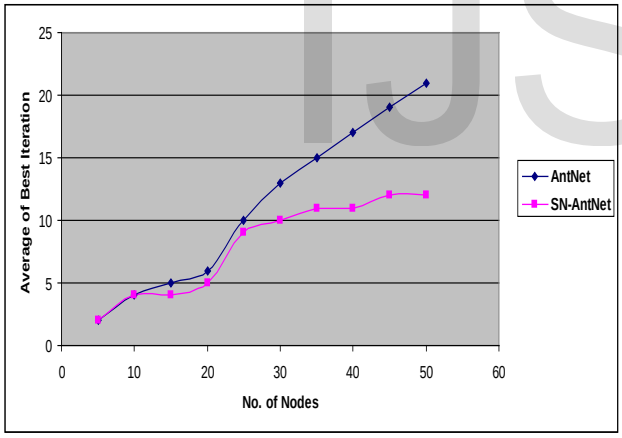

Fig.10: Chart of Best Iteration Where Number of Nodes of Maximum Ant Numbers is Equal to half of the Number of Nodes.

Figure (11) illustrates the curves of comparison between the original AntNet and the proposed Subset AntNet by investigating the relation between the number of nodes and the best number of Ants when the number of Ant is three quarter number of nodes. It is clear when the number of nodes is dropped between (5- 15) the Ant number is increased linearly with the number of nodes, and the two algorithms are comparable. In the modified algorithms best results appear clearly when the intermediate nodes between the source and destination are more than three nodes. the number of nodes is equal or exceed 20 we note that: For both algorithms the number of ants are continuous to increase because the number of nodes is increased hence more Ants are needed to find the destination, also because of the pheromone trail, many ants may be followed the same path before finding the best Ant that find the path with minimum cost. For the Subset Nodes Based AntNet again the number of ants needed is less than the number of ants needed in the original method. With reduced gap between the two AntNet Algorithms.

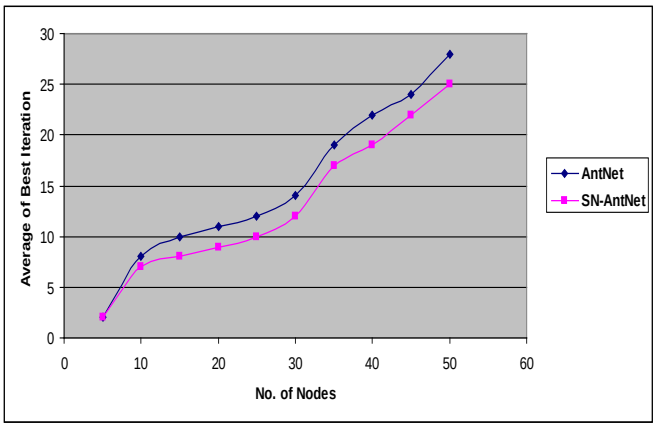

Fig.11 : Chart of Best Iteration Where Number of Nodes of Maximum Ant Number is Equal to Three Quarter of the Number of Nodes.

7 Conclusion

In this work, the Original AntNet has been tested in comparison with modified algorithm (Subset based Ant Net). AntNet is an algorithm used for packet routing in communication networks, a group of mobile agents build paths between pairs of nodes exploring the network concurrently and exchanging data to update the routing tables. This work shows a good performance of the AntNet in solving routing problems also the proposed algorithm offer an improvement on the original method by minimizing the iteration number and minimizing the ant number. In the modified algorithms best results appear clearly when the intermediate nodes between the source and destination are more than three nodes. The number of iterations and number of ants increased with increasing the number of nodes, therefore the proposed algorithms work on reducing the number of iteration and the number of ants. From the execution of Original AntNet and the Subset Nodes Based AntNet, it is clear that each one of them depends on certain parameters which are different from one to another; the results that give the best results are listed below:
For original AntNet: α=0.5, q0 = 0.9, r = 0.5
For the Subset Nodes Based AntNet: pm = 0.1,
From the experimental results it's clear that the Subset Nodes Based AntNet is the best in which gives good results and performance, in term of the following: Minimum path cost, Minimum iterations, and Minimum ant number.

References

[1] R.Geetha, G.Umarani Srikanth, "Ant Colony optimization based Routing in various Networking Domains – A Survey", International Research Journal of Mobile and Wireless Communications, Vol .03, Issue 01; February 2012, pp.424 -428. ISSN: 2249 – 6491.

[2] Sumit Sharma, Nikhil Marriwala, C.C.Tripathi," A Review on Different Routing Protocols in Ad-Hoc Network", INTERNATIONAL JOURNAL OF ADVANCES IN COMPUTING AND INFORMATION TECHNOLOGY, February 2012, ISSN 2277 – 9140.

[3] Chien- chung Shen, Chaiporn Jaikaeo, "Ad hoc multicast routing algorithm with swarm intelligence", Published in Mobile Network and Applications, Volume 10 Issue 1-2, February 2005.

[4] Benjamín Barán, Rubén Sosa, "AntNet: Routing Algorithm for Data Networks based on Mobile Agents", Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. NO.12 (2001), pp 75-84. ISSN: 1137-3601.

[5] Gianni Di Caro, Marco Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks", Journal of Artificial Intelligence Research 9 (1998) 317-365 Submitted 5/98; published 12/98.

[6] F. Tekiner, Z. Ghassemlooy, "Improved Antnet Routing Algorithm for Packet Switching", Mediterranean Journal of Computer Networks, Oct 2005, Vol 1, No,2, pp. 69-76.

[7] Majid Yousefi khoshbakht, Mohammad Sedighpour, "An Optimization Algorithm for the Capacitated Vehicle Routing Problem Based on Ant Colony System", Australian Journal of Basic and Applied Sciences, 5(12): 2729-2737, 2011 ISSN 1991-8178.

[8] S.S. Dhillon* , P. Van Mieghem," Performance analysis of the AntNet algorithm ", Computer Networks 51 (2007) 2104–2125.

[9] Gianni Di Caro, "Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks " IRIDIA, Institut de Recherches Interdisciplinaires et de D´eveloppements en Intelligence Artificielle, 2004.

Raghad Zuhair Yousif received BSc in Electronics and Communication Engineering from Baghdad University College of Engineering Department of Electronics and Communication Engineering in 1998. Then he received MSc. In Electronics and Communication Engineering from Al-Mustansriyha University in Baghdad College of Engineering Department of Electrical Engineering in 2001. His research was in field of image processing and data security. Then he received a PhD.in Communication Engineering form Department of Electrical and Electronic Engineering Baghdad University of Technology in 2006. His research was in field of FPGA and channel coding. He had been worked as senior lecturer at Department of Software Engineering College of Engineering Salahaddin University–Hawler from 2006 to 2010. He is currently Professor Assistant at branch of Communication in Department of Applied Physics College of science at Salahaddin university-Hawler. His research areas of interest are Reconfigurable hardware, Channel Coding, Real Time Systems, Medical Image processing, Remote sensing, Data Security, Bioinformatics and Computer Networks. He is senior lecturer at many MSc Courses for remote sensing, Advanced Computer Networks, Multimedia Technology, Network Security, Real time systems, and supervisor of Many MSc. Thesis in Software Engineering.