http://fbe.trakya.edu.tr/tujs
Trakya Univ J Sci, 9(1):25-29, 2008
ISSN 1305–6468
DIC: 239SZE910805080608

Araştırma Makalesi / Research Article

REFINDING OF THE OPTIMAL PATH WHEN THE NETWORK CHANGES

Şenol Zafer ERDOĞAN[*], E. Murat ESİN[*], Erdem UÇAR[**]

[*]Maltepe University, Faculty of Engineering, 34857 Maltepe Istanbul, Turkey
e-posta: {senole, emesin}@maltepe.edu.tr
[**]Trakya University, Faculty of Engineering and Architecture, 22100 Edirne, Turkey
e-posta: erdemucar@trakya.edu.tr

Alınış: 10 Temmuz 2007
Kabul Ediliş: 29 Nisan 2008

Abstract: *In this study, a new intelligent agent type which tries to find the new optimal path when the network changes is used. Self Cloning Ant Colony Approach is used to find the optimal path. The approach focusing on the search and routing algorithm has been developed. Self Cloning Ants can clone or destroy themselves. These ants find the optimal path between the source and destination node. In this study, in a real network, two different network environments have been created and two routing tables in these different network environments have been updated by using this approach.*
Keywords: *Routing Algorithms, Ant Colony, Computer Networks*

Ağ Değişimlerinde Optimal Yolun Bulunması

Özet: *Bu çalışmada, yeni bir akıllı karınca tipi kullanılmaktadır ve onlar ağ değişimlerinde yeni optimal yolu bulmaya çalışmaktadırlar. Kendini klonlayan karınca kolonisi yaklaşımı optimal yolu bulmak için kullanılır. Bu yaklaşım, arama ve yönlendirme algoritmalarına odaklanarak tasarlanmıştır. Kendini klonlayan karıncalar kendilerini klonlayabilir ya da yok edebilirler. Bu karıncalar kaynak ve hedef düğüm arasındaki optimal yolu bulurlar. Bu çalışmada, gerçek bir ağ içerisinde, iki farklı ağ ortamı oluşturulmuştur ve bu farklı ağ ortamlarındaki iki yönlendirme tablosu bu yaklaşım kullanılarak güncellenmiştir.*
Anahtar Kelimeler: *Yönlendirme Algoritmaları, Karınca Kolonisi, Bilgisayar Ağları*

1. Introduction

Routing strategies and protocols are important for the network information systems and telecommunication. Routing in a network is the action of addressing data traffic between pair of nodes (source-destination), being this, fundamental in a communication network control [1][2][3]. In network environment, routers take the decisions using routing tables. Routing strategies currently in widespread use (e.g. OSPF, RIP and BGP) are implemented through the information contained in routing tables independently available at each node in the network [4][5]. That is, the table consists of specific entries for the neighboring nodes and then a series of defaults paths for packets with any other destination – for example OSPF or BGP4 [6][7]. Many studies have been carried out for taking routing decision and several routing algorithms.

Nowadays, mobile agents are used to take routing decisions and to obtain some information on network. Well known mobile agent routing algorithm is Antnet [8]. The Antnet algorithm is based on the ant colony system (ACS) concept. ACS is an optimization method where a group of artificial ants move around a graph depending on the problem that exists; so, they move on the graph for building solutions and modifying the problem using the obtained information, until finding appropriate solutions to the problem [1][2].

The other mobile agent routing algorithm is Self Cloning Ant colony approach. Self Cloning Ant colony approach is used in the search and routing algorithms. In this approach, a synthetic ant called as Self
Cloning Ant is used. These ants may assess the situation and then they multiply through cloning or destroy themselves [2]. These ants find the optimal path between the pair of nodes in a real net.

2. Self Cloning Ant Colony Approach

Self cloning ant colony approach focusing on the search and routing algorithms has been developed [2][3][9]. This approach is used to find the optimal path between a pair of nodes in an unknown net topology. In this approach, a synthetic ant type is defined known as Self Cloning Ant.

In nature, it is impossible for ants to know completely their transportation network. The ants leaving their nests will follow a path randomly. These real ants use chemicals called pheromones to mark the paths they have taken. An ant that perceives a pheromone deposited by another individual of the same species may deduce that the species has reached that point previously.

In this approach, self cloning ants are used and they don't need to know the entire network. Self cloning ant does not have any information about network topology. The network mentioned in this paper consists of the nodes. Each node is a part of the network and each node must establish one connection with another node. There are some data structures in this approach. One of them is the Neighborhood list. Every node has a neighborhood list. This list records the neighboring nodes and their connection ports. Figure 1 shows the simple net and Table 1 shows the neighborhood list of each node. Second structure in this approach is Pheromone Status Flag (PSF). Each node has this flag to describe that this node had already been visited by any ant which has a similar task just like real ants leaving pheromone behind [2][3][9]. If PSF's status is as "visited", it means this node has been visited by the other self cloning ant previously. The last structure is the path list. Every ant has a path list. The path list is a list which includes nodes' name that is being visited by an ant. In the approach, hop count and expiration time which were used in Perkins and Royer [11] aren't needed to keep in the structure. Because, due to nature of our approach, finding of destination node is desired, if exist in the net. If not, every clone already will be destroyed at any node visited before. This situation eliminates the possibility of the traffic which makes busy the net forever.

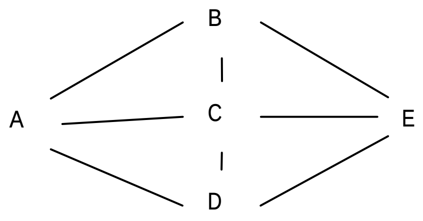

Figure 1. The simple net

Table 1. The First Structure: The neighborhood lists (1 means existence of connection between these two nodes, 0 means no existence between these two nodes)

	A	B	C	D	E
A	0	1	1	1	0
B	1	0	1	0	1
C	1	1	0	1	1
D	1	0	1	0	1
E	0	1	1	1	0

2.1. Process of the Self Cloning Ant Colony Approach

The aim of the self cloning ant is to reach the destination node by using the optimal path. At the start of the process, the source node creates a self cloning ant. The source node's name is added to the ant's path list. The self cloning ant scans the node's neighborhood list and clones itself for each link that the node has and then every cloning ant travels on separate links. The pheromone status flag is checked on the node that is being visited by the self cloning ant. If flag's status is as "visited", it is apparent that the path taken by an ant is not the shortest one. So it will only cause unnecessary traffic; therefore the most appropriate solution for the ant is to destroy itself. If flag's status is not as "visited", the ant reads the node's name and adds the name of the node arrived to the path list. After that, it compares the node's name with the name of the destination node. If the match is true, the ant changes flag's status as "visited" and it returns to source node. If the match is false, ant changes flag's status as "visited" and it clones itself as many as the number of connections that the node has, after this, each cloning ant travels to separate links [2].

The behavior model of the self cloning ant is described below as an algorithm [2].

```
1.  Read the pheromone status flag (PSF) of the node arrived.
2.  If PSF equal to" visited" then go to 11.
3.  Read the node's name.
4.  Add the name of the node arrived to the path list.
5.  Add the distance between the previous node and the node arrived to the path list.
6.  If node's name equal to destination node's name then change flag's status as "visited" and return to source
    node.
7.  Read the node's neighbors list except the name of the entering port.
8.  Produce clones as many as the number of connections to dedicate a port for each clone.
9.  Access to the dedicated port to reach to a new node.
10. Go to step 1.
11. End.
```

The path list of the ant which reaches the destination node is the optimal path list. The other cloning ants will destroy themselves in a period of time.

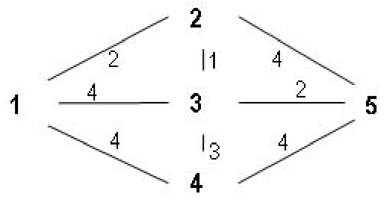

Figure 2. Nodes and distances information

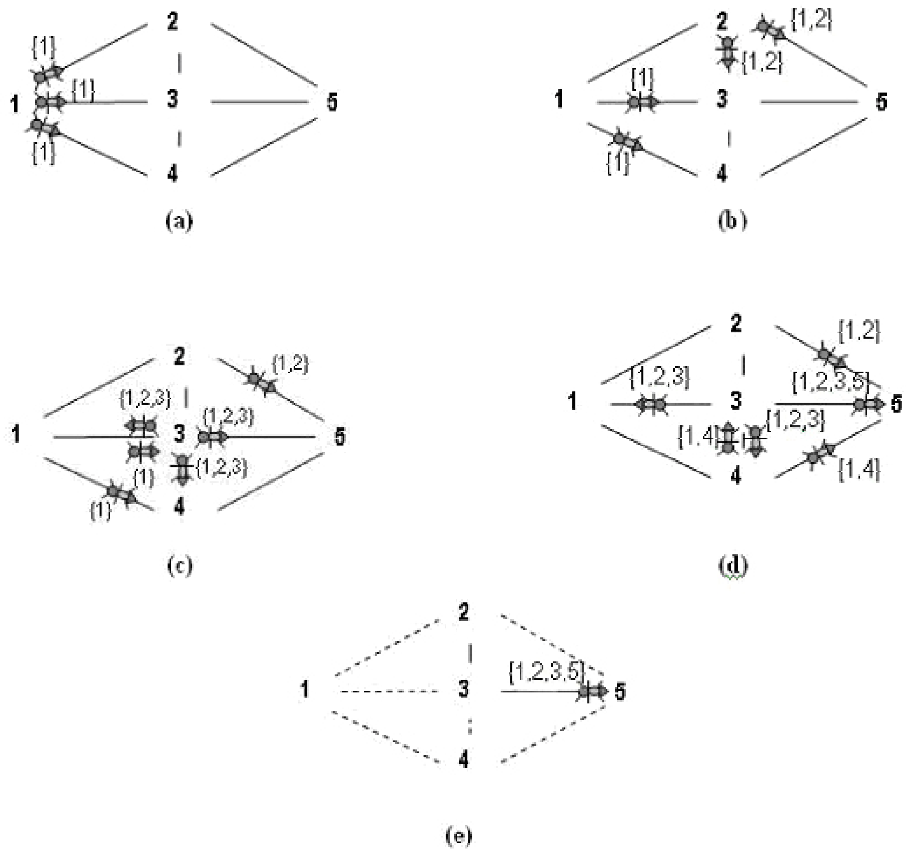

Figure 3. The process of the approach

In figure 3.a., at the start of the process, node 1 creates a self cloning ant. This ant adds the source node's name to its path list. After that it scans the neighborhood list and clones itself as many as the number of the connections that the node has. Every ant travels on separate links. After 2 units time in figure 3.b., the ant that uses (1-2) path, arrives at the node 2. This ant checks the Pheromone Status Flag (PSF). If Node's PSF is as "visited" then it destroys itself. If flag is not as "visited", it changes the node's flag status as "visited" and it adds the name of the node arrived to the path list. If this node is destination node then it returns to the source node. If node is not destination node, the ant clones itself as many as the number of the connections that the node has. The ants that travel (1-3) and (1-4) paths continue to travel along these paths. After 3 units time in figure 3.c., the ant that uses (2-3) path, arrives at the node 3 and applies the same steps as explained above. After 4 units time in figure 3.d., the ants that travel (1-3) and (1-4) paths arrive at the node 3 and node 4. At node 3, ant checks the node's flag status and sees the node's flag as "visited" and ant destroys itself. At node 4, ant checks the PSF. It adds the name of the node 4 and it clones itself as many as the number of the connections that node 4 has. After 5 units time in figure 3.e., the ant that travels (3-5) path, arrives at the node5 that is the destination node. The ant changes flag's status as "visited" and it adds the name of the node to the path list. It determines that it is destination node and then it returns to the source node. Since the destination node is already been reached by an ant, the other self cloning ants destroy themselves in a period of time. As a result, only one self cloning ant arrives at the destination node and this ant has the optimal path from source node to destination node.

3. Application

All experiments were implemented with the OMNET++ version 3.1 [10]. OMNET++ is an object-oriented modular discrete event simulation system. We used a real net topology, NSFNET, the USA backbone. It is composed of 14 nodes and 21 bi-directional links. The topology and the propagation delays are shown in figure 4. The bandwidth is 1.5 Mbit / s for every link, the assigned link or node fault probability is null, local buffers have a gigabyte of capacity and links are accessed through statistical multiplexing.

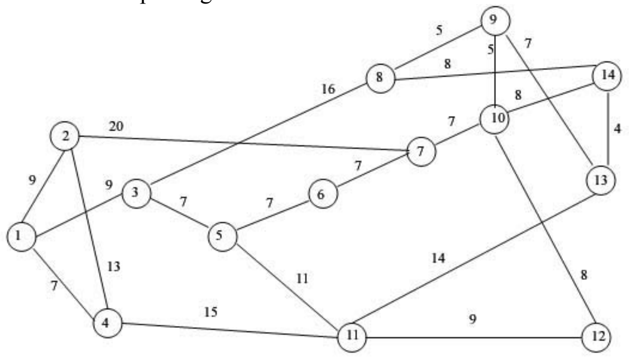

Figure 4. NSFNET (nodes and distances information)

In this section, two applications are explained. In the first application, we assume that all the nodes and the links work properly. In the second one, a node (node 8) and its links are destroyed or closed. In this situation, new paths are created by using Self Cloning Ant Colony Approach.

In the first application, node 1 is used as source node. NSFNET that is shown in figure 4 is used for simulation. In figure 4, nodes connections between the nodes and distances information in ms are shown. Every node is selected as destination node in sequence. For every node, the source node uses the Self Cloning Ant Colony Approach to find the optimal path between the source and destination node and creates a self cloning ant for each link. Self cloning ant travels through the net as explained in section 2 and for every node, an optimal path information is created. The first arrived self cloning ant at the destination node possesses the optimal path information. In Table 2, Node 1's routing table gives us the optimal path information for every node. When we consider the information given on Table 2, to arrive from node 1 to node 14, (1-3-8-14) path must be used. This path is optimal path between node 1 and node 14. Every row on the table shows the optimal path between the node 1 and other nodes.

Table 2. Node 1 routing table

Destination	Route List	Time
1	1	0 ms
2	1-2	9 ms
3	1-3	9 ms
4	1-4	7 ms
5	1-3-5	16 ms
6	1-3-5-6	23 ms
7	1-2-7	29 ms
8	1-3-8	25 ms
9	1-3-8-9	30 ms
10	1-3-8-9-10	35 ms
11	1-4-11	22 ms
12	1-4-11-12	31 ms
13	1-4-11-13	36 ms
14	1-3-8-14	33 ms

In the second application, node 8 and the connections with the node 8 are destroyed or closed. In this situation (1-3-8-14) path can not be used to arrive at the node 14 so data packets do not arrive at the node 14 and they go back. After that the source node creates a self cloning ant and sends it through the net and updates its own routing table. Every node is selected as the destination node. The process continues as explained in section 2 and at the end of the process node 1's routing table is updated and the optimal path information for node 14 is created. Updated routing table is shown at Table 3.

Table 3. Node 1 routing table after the network changed

Destination	Route List	Time
1	1	0 ms
2	1-2	9 ms
3	1-3	9 ms
4	1-4	7 ms
5	1-3-5	16 ms
6	1-3-5-6	23 ms
7	1-2-7	29 ms
8	-	-
9	1-2-7-10-9	41 ms
10	1-2-7-10	36 ms
11	1-4-11	22 ms
12	1-4-11-12	31 ms
13	1-4-11-13	36 ms
14	1-4-11-13-14	40 ms

In Table 3, the optimal path information and time information have changed for Node 9, Node 10 and Node 14. In the first application, it is seen that (1-3-8-14) path must be traveled to arrive from node 1 to node 14. In the second application, the optimal path from node 1 to node 14 is (1-4-11-13-14). In the first application, the elapsed time between node 1 and node 14 is 33 ms. and in the second application, is 40 ms.

4. Conclusion

In this study, Self Cloning Ant Colony Approach and an ant's behavior are explained. When the net topology changes dynamically, nodes create a self cloning ant to update their routing tables.

Self cloning ants assess the situation in a net and then they choose either cloning or destroying themselves. So we do not need to know how many ants must be used in a net topology. Self cloning ant clones itself for every link in a net topology. For this network mentioned in this study, 27 ants were cloned and they scanned the net and in the ant they destroyed themselves within a period of time.

There is an optimal path for every node that must establish at least one connection with another node. Every node has the optimal path information for every other node in its routing table. Every row in routing tables shows us the optimal path between the source node and the other node which is in net.

References

1. BARAN, B., SOSA, S. (2000), A New Approach for AntNet routing, Computer Communications and Networks, Proceedings Ninth International Conference, 303- 308

2. ESİN, E. M., ERDOGAN, Ş. Z. (2006) Self Cloning Ant Colony Approach and Optimal Path Finding, Proceedings of the Euro American Conference on Telematics and Information Systems (EATIS 2006), Colombia

3. ERDOGAN, Ş. Z., ESİN, E. M. (2006) Routing Table Updating by Using Self Cloning Ant Colony Approach, Proceedings of 5th International Symposium on Intelligent Manufacturing Systems (IMS'2006), Sakarya, Türkiye

4. YUN, H., ZİNCİR-HEYWOOD, A. N. (2004) Intelligent Ants for Adaptive Network Routing, Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'2004)

5. FOROUZAN, B. A. (2001) Data Communications and Networking, Mc-Graw Hill, ISBN:0-07-232204-7

6. HALABİ, B. (1997) Internet Routing Architectures, Cisco Press, ISBN: 1-56205-652-2,

7. LİANG, S., ZİNCİR-HEYWOOD, A. N., HEYWOOD, M. I. (2005) Adding More Intelligence to the Network Routing Problem: AntNet and GA-agents, Elsevier

8. CARO, G. D., DORIGO, M. (1998) AntNet: Distributed Stigmergetic Control for Communications Networks, J. Artificial Intelligence, vol. 9, 317-365

9. ERDOGAN, S. Z., ESİN, E. M. (2006) An Application of Two Different Algorithms on the Same Network, Proceedings of the International Symposium on Communications and Information Technologies 2006 (ISCIT2006), Thailand

10. OMNET++, http://www.omnetpp.org/external/whatis.php, June 2005

11. PERKINS, C. E., Royer, E. M., 1999, Ad-hoc on demand distance vector routing, In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pages 90 - 100, New Orleans