

# An Adaptive Multi-Agent Routing Algorithm Combining AntNet and Interconnected Learning Automata

Z. Farhadpour

Department of Computer Engineering  
Islamic Azad University  
Farahan Branch, Iran  
z.farhadpour@gmail.com

M. R. Meybodi

Department of Computer Engineering  
Amirkabir University  
Tehran, Iran  
mmeybodi@aut.ac.ir

## Abstract

*Learning Automata (LA) is an abstract model which can be used to guide action selection at any stage of a system by past actions and environment responses to improve some overall performance function. The use of intelligent algorithms based on learning automata can be efficient for traffic control. However, these learning schemes have been focused only to unimodal routing problem in connection oriented networks. The field of Ant Colony Optimization (ACO) models real ant colony behavior using artificial ant algorithms and find its application in a whole range of optimization problems. Ant algorithms experimentally prove to work very well in static and dynamic optimization problems and match perfectly with some model of interconnected LA. In this paper, an adaptive multi-agent routing algorithm called LA-AntNet is proposed for both source and non-source routing in communication networks. In this algorithm, mobile ant agents form AntNet routing system (Dicaro & Dorigo 1998) are combined to a system of static distributed LA agents, statically connected to the network nodes and directly responsible for routing decisions. The mobile ant agents improve local decisions of LA and adapt it with network conditions by moving over the network and collecting information about traffic distribution. In this algorithm the decision policy of LA is modified to involve a heuristic parameter which is a suggestion coming from ACO field and can guide learning process and improve convergence results. The proposed algorithm is implemented on several topologies to obtain performance metrics namely, throughput and total delay. The results are compared to the ones obtained from AntNet and a learning automata technique.*

**Keywords** : Routing, Learning Automata, AntNet , Algorithm

## 1. Introduction

LA have been traditionally used to model biological learning systems and to learn an optimal action that a Random Environment (RE) offers. Learning is achieved by interacting with the environment, and processing its responses according to the chosen actions. In the learning process an automaton is presented with a set of actions by the environment with which interacts, and it chooses one of these actions. Based on the chosen action, the automaton is either rewarded or penalized by the environment with a certain probability. Based on this response, the automaton attempt to learn the optimal action. [7]. Learning is not only considered in the single automaton case, but also hierarchies of LA and distributed interconnection of LA can be used for modeling and controlling complex environment such as communication networks.

The filed of ACO, which is inspired by the behavior of real ant colonies, studies artificial systems, used for discrete optimization problems. The main observation on which ACO is based is that real ants are capable of finding shortest paths from their nest to food sources and back. They can perform this behavior thanks to a simple pheromone laying mechanism. When ants move from their nest to the food source they move mostly random, but their random movements are biased by pheromone trail left on the ground by preceding ants. Because the ants that initially choose the shortest path to the food arrive first, this path will be seen as more desirable by some ants during their journey back to the nest. This, in turn, will increase the amount of pheromone deposited on the shortest path. Eventually, this auto-catalytic process causes all the ants to take the shortest path. Artificial ants take advantage of the differential length as well as of the auto-catalytic aspects of the real ant's behavior to solve discrete optimization problems. The problem description is represented by a graph. Artificial ants are software

agents in this graph, who modify some variable so to favor the emergence of good solutions. In practice, to each graph's edge a variable is associated, which is called a pheromone trail in analogy with the real ants. Ants add pheromone to those edges they pass and by doing so they increase the probability with which future ants will take these edges. Artificial ants, as real ones, move according to a probabilistic decision policy biased by the amount of pheromone trail they smell on the graph edges. In this paper an adaptive multi-agent routing algorithm called LA-AntNet is proposed in which mobile ant agents from ant algorithms are combined to a system of static distributed LA agents, statically connected to the network nodes and directly responsible for routing decisions. The mobile ant agents improve local decision policy of LA agents and adapt it with network conditions by moving over the network and collecting information about traffic distribution. In addition, the local decision policy of learning automata is modified to involve a heuristic parameter which is a suggestion coming from ACO field which can guide the learning process and improve convergence results. The rest of this paper is organized as follows:

Section II shows the principle of LA, in section III ant algorithms are discussed, section IV shows the data structure and the network model of the proposed algorithm and describes the logic of this algorithm, in section V the performance of the proposed algorithm is evaluated and the simulation results are reported. Section VI is the conclusion.

## 2. Learning Automata

A LA formalizes a general stochastic system in terms of states, actions, state or action probabilities and environment responses [7]. The design objective of an automaton is to guide the action selection at any stage of a system by past action and environment responses, so that some overall performance function is improved. At each stage, the automaton chooses a specific action from its finite action set and the environment provides a random response (see fig. 1).

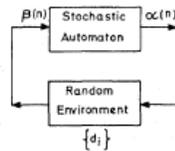


Fig. 1. Learning automata—environment pair.

A variable structure stochastic automaton, is defined by a quadruple  $\{\alpha, \beta, P, T\}$  for which  $\alpha$  is the action or output set  $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$  of the automaton,  $\beta$  is

random variable in the interval  $[0,1]$ ,  $P$  is the action probability vector of the automaton or agent and  $T$  denotes an update scheme. The output of the automaton is actually the input to the environment. The input  $\beta$  of the automaton is the output of the environment, which is modeled through penalty probabilities  $c_i$ , with  $c_i = P[\beta | \alpha_i], i: 1, \dots, r$ .

Important examples of linear update schemes are linear reward-penalty, linear reward-inaction and linear reward- $\epsilon$ -penalty. The general reward- penalty algorithm is given by:

$$P_i(n+1) = P_i(n) + a(1 - \beta(n))(1 - P_i(n)) - b\beta(n)P_i(n) \quad \text{if } \alpha \text{ is chosen at tim} \quad (1)$$

$$P_j(n+1) = P_j(n) - a(1 - \beta(n))P_j(n) + b\beta(n)\left[(r-1)^{-1} - \right]P_j(n) \quad \text{if } \alpha_j \neq \alpha_i \quad (2)$$

The constants  $a$  and  $b$  are the reward and penalty parameters, respectively. When  $a=b$ , the algorithm is referred to as linear reward- penalty ( $L_{R-P}$ ); when  $b=0$ , it is referred to as reward-inaction ( $L_{R-I}$ ); and when  $b$  is small compared to  $a$ , it is called linear reward- $\epsilon$ -penalty ( $L_{R-\epsilon P}$ ).

## 3. ACO and Routing in Communication networks

The field of ACO studies artificial Ant systems for solving discrete optimization problems. The problem description is represented in a graph and the artificial ants are software agents who can move in this graph to find out one or more shortest path. The construction of shortest path is motivated by how real ants find the shortest path from their nest to a food source and back. By analogy with the pheromone trail real ants use, each graph edge  $(i, j)$ , connecting node  $i$  with node  $j$ , has a variable  $\tau_{ij}$  associated with it. This variable represents the amount of pheromone on the edge. Walking ants move according to a probabilistic decision policy based on the amount of pheromone trail they smell on the graph's edges. Positive feedback is thus improved by the reinforcing a trail on those edges which are used. To avoid some premature convergence this pheromone trail evaporated over time and ants, transitions to other nodes in the graph happen stochastically. Communication in this system happens locally and indirectly. Ants cooperate by leaving information of each other at the same places in the graph. Recently, a number of routing algorithms inspired by ant colony metaphor have been proposed for both wired and wireless networks. AntNet [5] is a routing algorithm proposed for wired datagram networks based on the

principle of ant colony optimization. In AntNet, each node maintains a routing table and an additional table containing statistics about the traffic distribution over the network. The routing table maintains for each destination and for each next hop a measure of the goodness of using the next hop to forward data packets to the destination. These goodness measures called pheromone variables are normalized to one in order to be used by a stochastic routing policy. AntNet used two sets of homogenous mobile agents called forward ants and backward ants. The forward ants use heuristic based on the routing tables to move between a given pair of nodes and are used to collect information about the traffic distribution over the network. The backward ants relate the paths of forward ants in the opposite direction. At each node, the backward ants update the routing table and the additional table containing statistics about the traffic distribution over the network.

#### 4. LA-AntNet Algorithm

In this section an adaptive multi agent routing algorithm called LA-AntNet is proposed which can be used for both source and non-source routing in communication networks. LA-AntNet consists of two set of non-homogenous RL agents who are trying to achieve a common goal(to adaptively learn routing tables).In this algorithm, mobile ant agents derived from the routing algorithm AntNet [4] ,[5] are combined to a network of interconnected LA. The AntNet agents (ants) are mobile distributed agents that collect information to support the operations of LA agents statically connected to the network nodes and directly responsible for routing decisions. An ant can be viewed as a dummy mobile agent that walks around in the graph of interconnected LA, makes states / LA active, and brings information so that the LA involved can update their local state. In this algorithm at regular intervals, from every network node  $s$ , a forward ant  $F_{s \rightarrow d}$  is launched with a randomly selected destination node  $d$ . Arriving at each node the forward ant will activate the automata stated in that node in order to determine the next hop node. Since several ants, are launched simultaneously, several LA can be active at the same time. However, adding multiple mobile agents to the system will not harm the convergence because no competition is involved. This should also be reflected in the fact that when using more ants, good solutions will evolve more quickly. [6].In LA-based systems every active LA chooses an action only based on its action probability vector. The experiments show that the results of such systems are almost not acceptable being far from optimum results. Also, the convergence rate of these algorithms is very

low. According to this, in the proposed algorithm the decision policy of LA is modified to involve a heuristic parameter,  $l_n$ , in addition to the  $P_{nd}$  value stored in action probability vector of active LA. This Select-Link function is shown in fig.2. In this function, the heuristic correction  $l_n$  is a  $[0, 1]$  normalized value proportion to the length  $q_n$  (in bits waiting to be sent) of the queue of the link connecting the node  $k$  with its neighbor  $n$  and is calculated as (3):

$$l_n = 1 - \frac{q_n}{\sum_{n=1}^{|N_k|} q_n}, \quad (3)$$

$l_n$  Gives a quantitative measure associated with the queue waiting time. The value of  $\alpha$ , in Select-Link function, weights the importance of the heuristic correction with respect to the probability value stored in the routing tables. In this way agent's decisions are taken on the basis of a combination of a long-term learning process and an instantaneous heuristic prediction.

As shown in fig.2, after selecting an action by LA the probability vector of its LA returns to its previous values. So that, the heuristic parameter is considered instantaneously in decision making function and does not affect the long-term learning process.

```

Select- Link (i){
// i is the current automata
// modify the action probability vector

$$P^i = \left\{ P_j^i \mid P_j^i = \frac{P_j^i + \alpha l_n}{1 + \alpha (|N_k| - 1)} : j = 1, 2, \dots, r \right\}$$

Choose an action as a sample realization of modified
action probability vector  $P^i$ .

Restore the previous value of  $P^i$ .
}

```

Fig.2 Select-link function

At the rest of this section we will first describe the network model and the data structures of the algorithm, and then we will discuss how the algorithm works.

#### 4.1. Network Model

The network is modeled as a graph  $G = (V, E)$  consisting of  $V$  nodes and  $E$  bidirectional links. In this network we state a variable structure LA for every possible destination node  $d$  at every node of the graph. At every instance its task is to choose a suitable edge form all outgoing edges in that node to reach final

destination  $d$ . the output is a sequence of chosen actions by LA which indicates a tour on the graph, which will serve as the input to the environment. Every node in the network maintain an input buffer composed of a single queue and an output buffer composed of high priority queue and a low priority queue for each neighbor or outgoing link. All the packets within the network can derive into two different classes:

- Data packets: Represent the information that the end users exchange with each other. These packets use the information stored at routing tables for traveling from the source to destination node.
- Ant packets (mobile ant agents): these packets are consist of two homogenous mobile agents which are forward and backward ants. These packets are used to collect and distribute information about the traffic load in the network or for updating local routing tables.

## 4.2. Data structures

Two data structures stored at each network node  $k$ :

- A routing table  $T_k$ , organized as a matrix with probabilities entries. Each row in the routing table corresponds to one destination in the network and each column corresponds to a neighbor of the current node. Each row of the table indicates the probability vector of a LA stated in the node. the routing table  $T_k$  store the probability vector of all LA stated in node  $k$  and indicate the probability of choosing neighbor  $n$  as the next node when the destination is  $d$ , such that for each LA:

$$\sum_{n \in N_k} P_{nd} = 1 \quad (4)$$

- A table  $\mu_k(\mu_d, \sigma_d^2, w_d)$ , containing statistics about the network topology and the traffic distribution over the network as seen by the local node  $k$ .

## 4.3. How does the algorithm work?

1. At regular intervals, from every network node  $s$ , a mobile ant agent called forward ant ( $F_{s \rightarrow d}$ ) is launched with a randomly selected destination node  $d$  to discover a feasible, low -cost path to that node and to investigate the load status of the network. To start, every forward ant will activate LA stated in its corresponding node to choose next hop node. In this way, several LA are activated simultaneously in the network and searching for shortest path is started.

2. Every active LA chooses an action from its modified probability vector as shown in *Select-Link* function.

Implementing this action will transfer the forward ant to the next hop node and will activate the LA stated in the other side of the link. This process continues until the forward ant reaches the destination  $d$ .

3. When the destination node  $d$  is reached, the forward ant  $F_{s \rightarrow d}$  generates another agent called backward ant  $B_{s \rightarrow d}$  transfers to it all of its memory, and dies.

4. The backward ant takes the same path as that of its corresponding forward ant, but in the opposite direction. At each node  $k$  along the path it pops its stack  $S_{s \rightarrow d}(k)$  to know the next hop node.

5. Arriving at a node  $k$  coming from a neighbor node  $f$ , backward ant updates the two main data structures of the node, the local model of the traffic  $\mu_k$  and the routing table  $T_k$ , which is containing the probability vector of LA for all the possible destination nodes.

The mean  $\mu_d$  and variance  $\sigma_d^2$  entries in the local traffic model  $\mu_k$  are modified using (5), (6).

$$\mu_d \leftarrow \mu_d + \eta(t_{k \rightarrow d} - \mu_d) \quad (5)$$

$$\sigma_d^2 \leftarrow \sigma_d^2 + \eta((t_{k \rightarrow d} - \mu_d)^2 - \sigma_d^2) \quad (6)$$

$t_{k \rightarrow d}$ , represents the newly observed forward ant's trip time to travel form node  $k$  to destination node  $d$  and  $\eta \in [0,1]$  is a factor that weights the number of recent samples that will affect the mean  $\mu_d$  and the variance  $\sigma_d^2$ .

The routing table  $T_k$  is changed by incrementing the probabilities  $P_{hd'}$  and decrementing, by normalization the other probabilities  $P_{nd'}$  as follows:

$$P_{hd} \leftarrow P_{hd} + r(1 - P_{hd}) \quad (7)$$

$$P_{nd'} \leftarrow P_{nd'} - rP_{nd'} \quad \forall n \neq h, n \in N_k \quad (8)$$

The reinforcement value  $r$  used in (7), (8) is a dimensionless constant  $(0, 1]$  and is calculated as:

$$r = c_1 \frac{t_{best d}}{t_{k \rightarrow d}} + c_2 \frac{t_{sup} - t_{best d}}{(t_{sup} - t_{best d}) + (t_{k \rightarrow d} - t_{best d})} \quad (9)$$

In (9),  $t_{k \rightarrow d}$  is the newly observed forward ant's trip time form node  $k$  to destination  $d$  and  $t_{best d}$  is the best trip time experienced by the forward ants traveling towards the destination  $d$  over the observation window  $w_d$ .

## 5. LA-AntNet performance evaluation

Several experiments were designed to evaluate the performance of the algorithm on several graph topologies. The results are compared with AntNet

routing algorithm and with a LA technique derived from LA-AntNet by ignoring heuristic parameter,  $I_n$ . Two performance metrics which were used to evaluate the algorithm are, throughput and total delay. The results of two sets of experiments are summarized below:

- Experiment set1: This experiment was implemented on a network with a regular topology consisting of 36 nodes. All the links in the network have bandwidth of 10 Mbit/s and propagation delay of 1 msec. The results are shown in fig 3, 4.
- Experiment set2: In this experiment, two sets of randomly generated networks of respectively 100 and 150 nodes are considered. The level of connectivity of each node has been forced to range between 2 and 5. Every link has the same bandwidth of 1 Mbit/sec, while the propagation delays have been generated in a uniform random way over the interval [0.01, 0.001]. The results are shown in fig 5, 6.

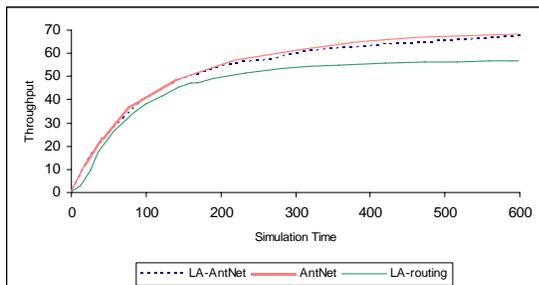


Fig.3. Throughput of the algorithms in experiment 1.

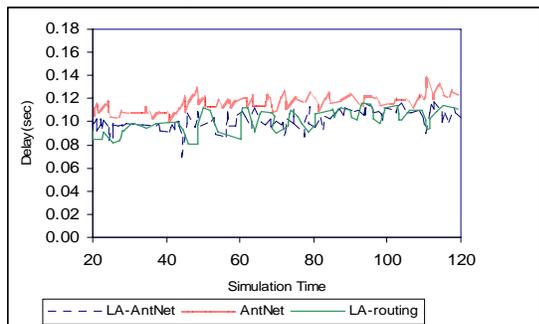


Fig.4. Average of End-to-End Delay in experiment 1.

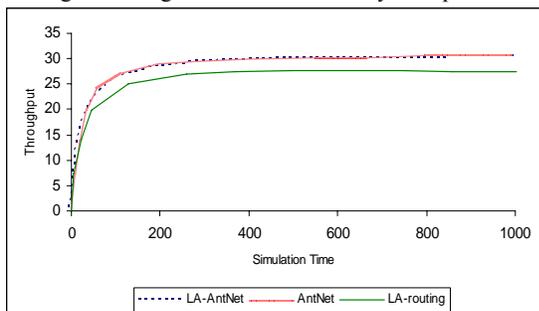


Fig.5. Throughput of the algorithms in experiment 2.

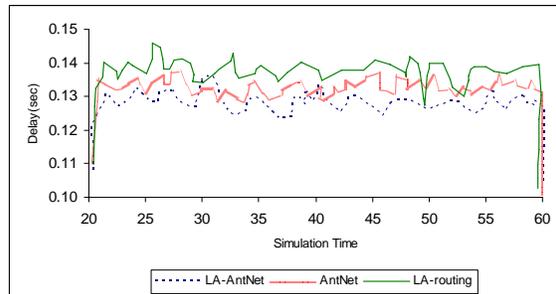


Fig.6. Average of End-to-End Delay in experiment 2.

The results show that the total delay of LA-AntNet is improved compared with LA technique and AntNet algorithm. On the other hand the throughput of LA-Antnet is as well as that obtained from Antnet algorithm.

## 6. Conclusions

In this paper, LA-AntNet algorithm is proposed, in which mobile ant agents form AntNet routing system are combined to a system of static distributed LA agents. Also, the decision policy of LA is modified to involve a heuristic parameter. The results show that the total delay of the algorithm is improved compared with LA technique and AntNet algorithm. On the other hand the throughput of LA-Antnet is as well as that obtained from Antnet algorithm.

## 7. References

- [1] F. Ducatelle, Gianni Di Caro, L.M. Gambardella, "Ant agents for Hybrid Multipath Routing in Mobile Adhoc Networks", 2nd Annual Conference On Wireless On demand Network Systems and Services (WONS), 2005.
- [2] S. Misra, B. J. Oommen, "Dynamic Algorithms for the Shortest Path Routing Problem: Learning Automata-Based Solutions", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 35, NO. 6, Dec. 2005.
- [3] K. Verbeeck, A. Nowé, "Colonies of Learning Automata", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 32, No. 6, Dec. 2002
- [4] Gianni Di Caro, Marco Dorigo, "AntNet: Distributed Stigmergetic Control for Communication Networks", Journal of Artificial Intelligence Research, vol. 9, pp. 317–365.
- [5] Gianni Di Caro, Marco Dorigo, "Two Ant Colony Algorithms for Best-effort Routing in Datagram Networks", International conference on parallel and distributed computing and systems (PDCS'98), pages 541-546, 1998.
- [6] M. Dorigo, V. Maniezzo, A. Colomi, "The Ant System: Optimization by a Colony of Cooperating Agents", IEEE Trans. Syst., Man, Cybern. B, vol. 26, pp. 29–41, 1996.
- [7] K. Narendra, M. Thathachar, "Learning Automata: An Introduction", Englewood Cliffs, NJ: Prentice-Hall, 1989.