Ergodic Discretized Estimator Learning Automata with high accuracy and high adaptation rate for nonstationary environments

A.V. VASILAKOS and G.I. PAPADIMITRIOU

Department of Computer Engineering
University of Patras
26500 Patras, Greece

ABSTRACT

In this paper a new ergodic discretized learning automaton which is epsilon-optimal is introduced. It utilizes a new estimator learning algorithm which is based on the recent history of the environmental responses and is able to operate in nonstationary stochastic environments. The proposed automaton achieves a significantly higher performance than the classic reward-penalty ergodic schemes. Extensive simulation results indicate the superiority of the proposed scheme. Furthermore, it is proved that it is epsilon-optimal in every stochastic environment.

I. INTRODUCTION

Learning Automaton is a finite state machine that interacts with a stochastic environment trying to learn the optimal action this environment offers. Over the last few years, learning automata have been extensively studied in the literature. We refer the reader to the outstanding survey papers by Narendra and Thathachar [6] and Narendra and Lakshmivarahan [7] for a review of various families of learning automata.

The purpose of a learning automaton that interacts with a stochastic environment is to learn the optimal action offered by the environment, via a learning process. The learning process is the following. The automaton chooses one of the offered actions according to a probability vector, which at every instant contains the probability of choosing each action. The chosen action triggers the environment that responses with an answer (reward or penalty) according to the reward probability of the chosen action. The automaton takes into account this answer and modifies its state by means of a transition function. The new state of the automaton corresponds to a new probability vector given by a function, called output function. A learning automaton is one which learns the action that has the maximum probability to be rewarded by the environment and which ultimately chooses this action more frequently than other actions.

With respect to the character of the previously referred transition and output functions, learning automata can be classified into two main categories: fixed structure automata or variable structure automata. Fixed structure automaton is characterized one whose transition and output functions are time invariant. The reader can consult Tsetlin, Krylov and Krinski automata [10], [11] in order to study examples of fixed structure learning automata. Variable structure stochastic automaton is characterized one whose transition and output function are time variable. They are more powerful learning machines, so most of the research in learning automata area has involved this category of automata.

Another way to classify learning automata is according to their Markovian representation. By this view, Learning Automata are classified into two main categories: ergodic [1], [2], [12] or automata possesing absorbing barriers [1], [2], [4], [5], [6].The ergodic automata converge with a distribution independent of the initial state. If the reward probabilities of the actions are

not stable (nonstationary environment), ergodic automata are preferred. On the other hand, the automata with absorbing states, after a number of finite steps get "locked" (converge) into a specific state. Absorbing automata are preferred, when the reward probabilities of the actions are stable (stationary environment).

According to the values the action probabilities can take, a learning automaton can be characterized as a continuous or a discretized one. In the former case the action probabilities can take any value in the [0,1] interval. This class of learning automata suffers from slow convergence. In the latter, the action probabilities can take values from a finite set only. In other words the [0,1] interval is divided into finite number of subintervals. If these subintervals are all of equal length, the automaton is characterized as a linear one; if not, it is a nonlinear one. By discretizing the probability space there is an increase in the speed of automaton's convergence. For the continuous automaton's case, as time tends to infinity, the choice probability of the optimal action tends to unity but never becomes equal to unity. The discretization of the probability space wipes this disadvantage out. By nonlinearing the division of the [0,1] interval we achieve further improvement of the automaton's performance [3], [4]. Important results of the application of discretized and nonlinear learning automata to network traffic management problems can be found in [13] and [14].

In this paper we present a new ergodic discretized estimator learning automaton (EDEL) which is epsilon-optimal in every stochastic environment.

The proposed automaton utilizes a window technique in order to estimate the reward probabilities of the actions. If a window of size W is utilized then the automaton takes into account only the W last environmental responses to each action in order to estimate the current reward probability of each action. The utilization of a window leads to the ignorance of "old" environmental responses. So, the automaton becomes able to operate in a nonstationary stochastic environment. After each iteration the choice probability of the action that has the highest estimated reward probability is increased.

Simulation results indicate that the proposed EDEL learning automaton achieves a significantly higher performance than the classic reward-penalty ergodic learning automata.

The structure of this paper is as follows. The introduction to the mathematical model of learning automata in Section II is followed by a brief review of estimator learning algorithms in Section III. Section IV introduces the reader of the paper to the new EDEL scheme, expanding its main advantages and analysing the main features of the proposed scheme. In Section V we give the formal definition of EDEL, followed by the proof of its epsilon-optimality in Section VI. Extended simulation results that prove the superiority of EDEL's performance are presented in Section VII. Finally, a brief discussion of the proposed scheme and its possible applications closes the paper in Section VIII.

II. LEARNING AUTOMATA

A learning automaton is defined by the quintuple <A,B,Q,T,G> where:

$$A=\{a_1,a_2,..,a_r\}$$

is the set of the r actions ($2\leq r<\infty$) offered by the environment,

$$B=\{0,1\}$$

is the input set of the possible environmental responses,

$$Q$$

is the set of the possible states of the automaton,

$$T: Q \times A \times B \rightarrow Q$$

is the learning algorithm and finally,

$$G: Q \rightarrow [0,1]$$

is the output function. We note that Q(t) denotes the automaton's state at time instant t. Q(t) includes (or determines) a probability distribution, P(t), over the action set A.

P(t) is defined as :

$$P(t) = \{p_1(t),p_2(t)...,p_r(t)\}$$

where $p_i(t)$ denotes the probability of selecting action a_i at time instant t; (thus

$p_i(t)= Pr[a(t)=a_i]$). Obviously,

$$\sum_{i=1}^{r} p_i(t)=1 \ ,\forall t.$$

Automata that maintain estimates of the environmental characteristics include an estimator vector:

$$D'(t) = \{ d'_1(t),d'_2(t),..,d'_r(t) \}$$

into their state Q(t). This vector contains the estimated reward probabilities of the actions at every time instant t.

The environment in which the automaton operates is defined by the triple:

$$<A,D,B>$$

where A and B are as defined above and

$$D(t)=\{d_1(t),d_2(t)..,d_r(t)\}$$

is the set that contains the reward probabilities of the actions offered by the environment. Thus:

$$d_i(t)=Pr[b(t)=1|a(t)=a_i\in A]$$

The environment is characterized as a "stationary" one if the reward probabilities are time invariant and as a "nonstationary" one if they are time variant.

A learning automaton is "optimal" if:

$$\lim_{t\to\infty} p_b(t)=1$$

with probability 1, where a_b is the action that has the highest reward probability. A learning automaton is characterized as an "epsilon-optimal" one if there is an internal parameter N such that:

$$\lim_{N\to\infty} \lim_{t\to\infty} E[p_b(t)] = 1$$

III. ESTIMATOR ALGORITHMS

Estimator learning algorithms were introduced by Thathachar and Sastry [8]. They were an approach to improve the convergence rate of learning automata.

Traditional learning algorithms update the probability vector based directly on the environment's answer. If this answer is a reward then the automaton increases the probability of choosing this action (which caused the reward) at the next time instant.

A learning algorithm is characterized as an "estimator" one if it uses a running estimate of the reward probability of each action. The change of the probability of choosing an action is based on its current estimated reward probability rather than on the feedback of the environment. The environment determines the probability vector, not directly, but indirectly, determining the estimates of reward probabilities. Even when an action is rewarded, it is possible that the probability of choosing another action is increased. Usually, estimator algorithms increase the choice probability of the action that has the highest estimated reward probability.

Simulation results have shown the superiority of the estimator learning algorithms over the classic learning algorithms [5],[8].

In this paper we present a new ergodic discretized learning automaton which utilizes an estimator learning algorithm able to operate in a nonstationary stochastic environment.

IV. THE ERGODIC DISCRETIZED ESTIMATOR LEARNING AUTOMATON (EDEL)

All classic ergodic learning automata reported in the literature up to now have a very important disadvantage: their performance becomes unacceptable when they operate in a nonstationary environment whose best action has a low (less than 0.5) reward probability. This is due to the fact that all the (other) ergodic automata were based on the classic reward-penalty scheme. According to this scheme when an action is penalized by the environment its choice probability is decreased. When the classic reward-penalty scheme tries to select the "best" action more frequently, if its reward probability is below 0.5, this action is penalized more frequently than it is rewarded. So, its choice probability is reduced. Under these conditions, the choice probability of the "best" action cannot take extreme values (closed to unity).

It is obvious that under these conditions the automaton's performance becomes unacceptable. This has been a limiting factor in the application of a reward-penalty ergodic scheme to an environment that offers low highest reward probabilities. Unfortunately, lots of stochastic environments belong to this category.

The learning automaton we propose is an effort to wipe this disadvantage out, offering a learning automaton able to operate successfully in any stochastic environment with high accuracy and high adaptation rate.

The main idea is to increase -at any instant- the choice probability of the action that has the highest estimated reward probability among all the other actions, independently to the specific value of its reward probability.

The proposed automaton utilizes a window of size W in order to estimate the current reward probability of each action. The current estimated reward probability of each action is defined as the sum of the W last environmental responses (0 or 1) to this action, divided by W. It is obvious that the utilization of a window leads to the ignorance of "old" environmental responses. So, the automaton becomes able to operate in a nonstationary environment. If the automaton operates in a rapidly switching environment, then only very recent environmental responses must be taken into account; thus, a small window size W must be utilized. If the environment is slowly switching then the choice of a large window size is more efficient. After each iteration the choice probability of the action that has the highest estimated reward probability is increased. If more than one actions have the same estimated reward probability, then the choice probability of the action that was selected least recently is inreased.

We note, that the proposed EDEL learning automaton is a discretized one. Thus, if a linear output function is utilized, the choice probability of the "optimal" action is increased (in the 2-action case) by a constant quantity $\Delta=1/N$. Where N is the number of the automaton's states minus 1. As it was discussed in the introduction, the discretization of the probability space leads to an increase of the automaton's adaptation rate.

The proposed EDEL learning automaton is an ergodic one. Assume the worst case. Let Pr[select a_i]=1, (Pr[select a_j]=0,$j\neq i$) $w_i[k]=1$ for k=1,..,W and $w_j[k]=0$ for k=1,..,W. Where w_i,w_j are arrays of size W, that contain the W last environmental responses to actions a_i and a_j correspondingly. Then the automaton is "deblocked" from this state if and only if: W continuous responses to action a_i will be equal to 0. This happens with probability:

$$\Pr[\text{ deblocking }] = (1-d_i)^W$$

where d_i is the reward probability of action a_i. For example, if $d_i=0.2$ and W=3 then: $\Pr[\text{deblocking}] = (1-0.2)^3 = 0.512$ Thus, it has been proved that the EDEL learning automaton is an ergodic one, able to operate in a nonstationary environment.

V. FORMAL DEFINITION OF THE EDEL LEARNING AUTOMATON

The EDEL learning automaton is defined as a quintuple <A,B,Q,T,G> where -for the 2 action case- we have:

$$A = \{a_1, a_2\}$$

is the set of actions offered by the environment.

$$B = \{0,1\}$$

is the set of the environment's responses (1 symbolizes a REWARD and 0 a PENALTY response).

$$Q(t) = \{s(t), D'(t)\}$$

is the state of the automaton, where:

$$s(t) \in \{ 0,1,2,\ldots,N \}$$

where N is an even integer parameter, characteristic for a discretized learning automaton, called "resolution parameter". s(t) determines the choice probability of each action at time instant t. If the G output function is utilized, then:

$P_1(t) = \Pr[\text{select action } a_1 \text{ at time t}] = G(s(t))$

$P_2(t) = \Pr[\text{select action } a_2 \text{ at time t}] = 1-G(s(t))$

$$D'(t)=\{d_1'(t), d_2'(t)\}$$

is the estimator vector that contains the current estimated reward probabilities of the actions. For i=1,2 we have:

$$d_i'(t) = \frac{\left\{\begin{array}{l}\text{times that action } a_i \text{ was}\\ \text{REWARDED during the last W}\\ \text{times it was selected}\\ \text{up to time instant t.}\end{array}\right\}}{W} \qquad (1)$$

G: is the output function. It may be a linear or a nonlinear one. The output function,

given the automaton's state, determines the choice probabilities of the actions. Its role was discussed above.

T: is the learning algorithm discussed in Section IV. Suppose that action a_i is selected at time instant t. Thus, $a(t)=a_i$. The environment responses with an answer $\beta(t) = 0$ or 1. Then the automaton computes the new value of current estimated reward probability of action a_i as it is given by type (1). The current estimated reward probability of action a_j ($j \neq i$) remains invariant.

Now, the current "optimal" action $a_{opt}(t)$ is computed as follows:

if $d_i'(t) > d_j'(t)$ then $a_{opt}(t)=a_i$
else $a_{opt}(t)=a_j$

Finally, we compute the new automaton's state s(t+1) according to the following updating scheme:

```
if  0<s(t)<N  then
{
 if a_opt(t) = a_1 then  s(t+1) := s(t) + 1
 if a_opt(t) = a_2 then  s(t+1) := s(t) - 1
}
if s(t)=N then
{
 if a_opt(t) = a_1 then  s(t+1) := N
 if a_opt(t) = a_2 then  s(t+1) := N-1
}
if s(t)=0 then
{
 if a_opt(t) = a_2 then  s(t+1) := 0
 if a_opt(t) = a_1 then  s(t+1) := 1
}
```

If the linear output function G(i)=i/N is utilized then the probability updating scheme is the following:

```
if  0<P_1(t)<1  then
{
 if a_opt(t)=a_1 then P_1(t+1) := P_1(t) + 1/N
 if a_opt(t)=a_2 then P_1(t+1) := P_1(t) - 1/N
}
if P_1(t)=1 then
{
 if a_opt(t) = a_1 then  P_1(t+1) := 1
 if a_opt(t) = a_2 then  P_2(t+1) := 1 - 1/N
}
if P_1(t)=0 then
{
 if a_opt(t) = a_2 then  P_1(t+1) := 0
 if a_opt(t) = a_1 then  P_1(t+1) := 1/N
}
```

Obviously, $P_2(t) = 1 - P_1(t)$,$\forall t$.

The Pascal-like description of the learning algorithm is presented below. We note that the arrays w_i (i=1,2) are of size W and contain the last W environmental responses to action a_i .

```
PROCEDURE LEARNING;
REPEAT
  select an action a(t) according
  to the probability vector ;
  ( let a_i to be the selected action
    and a_j to be the other action )
  receive the feedback β(t)
  (0 or 1) from the environment;
  (* UPDATE THE WINDOW *)
  for k:=W downto 2 do  w_i[k]:=w_i[k-1];
  w_i[1]:=β(t);
  (* COMPUTE THE CURRENT ESTIMATION *)
                  W
                 Σ  w_i[k]
                k=1
  d_i'(t) := ------------
                 W
  (* UPDATE THE AUTOMATON'S STATE *)
  if d_i'(t) > d_j'(t)  then  a_opt(t) := a_i
                        else  a_opt(t) := a_j ;
if a_opt(t)=a_1 and s(t)<N then  s(t+1):=s(t)+1
else
if a_opt(t)=a_2 and s(t)>0 then s(t+1):=s(t)-1;
  P_1(t+1) := G( s(t) );
  P_2(t+1) := 1 - P_1(t+1);
UNTIL FALSE;
```

VI. PROOF OF EPSILON-OPTIMALITY

Theorem: In every stochastic environment the EDEL learning automaton is ϵ-optimal. In other words if a_i is the optimal action

$(d_i>d_j, i\neq j)$ then there is a finite integer number W_{min} such that if W is the window size of the EDEL learning automaton and $W\geq W_{min}$ then:

$$\lim_{N\to\infty}\lim_{t\to\infty} E[P_i(t)] = 1$$

where N is the resolution parameter of the EDEL learning automaton and $P_i(t)$= Pr[select action a_i at time instant t]

Proof: For simplicity we symbolize the k^{th} state of the automaton as k, for k=0,..,N. Thus, when we write s(t)=k we mean that the automaton is in the k^{th} state at time instant t.

Assume an EDEL learning automaton with a window size equal to W that operates in a stochastic environment that offers two actions a_1 and a_2 which have reward probabilities equal to d_1 and d_2 correspondingly. Without loss of generality assume that $d_1>d_2$; thus, action a_1 is the optimal one.

At a random time instant t, the probability that the current estimation of action's a_1 reward probability $d_1'(t)$ is higher than the current estimation of action's a_2 reward probability $d_2'(t)$ is:

$$\Pr[\, d_1'(t) > d_2'(t) \,] = \sum_{k=1}^{W}\left[\binom{W}{k} d_1^k(1-d_1)^{W-k}\sum_{m=1}^{k-1}\binom{W}{m} d_2^m(1-d_2)^{W-m}\right] \quad (2)$$

The probability that $d_1'(t) = d_2'(t)$ is:

$$\Pr[\, d_1'(t) = d_2'(t) \,] = \sum_{k=0}^{W}\binom{W}{k}^2 d_1^k\,(1-d_1)^{W-k}\, d_2^k\,(1-d_2)^{W-k} \quad (3)$$

The probability that $d_2'(t) > d_1'(t)$ is:

$$\Pr[\, d_2'(t) > d_1'(t) \,] = \sum_{k=1}^{W}\left[\binom{W}{k} d_2^k(1-d_2)^{W-k}\sum_{m=1}^{k-1}\binom{W}{m} d_1^m(1-d_1)^{W-m}\right] \quad (4)$$

Obviously,

$$\Pr[\, d_1'(t)>d_2'(t) \,] + \Pr[\, d_1'(t)=d_2'(t) \,] + \Pr[\, d_2'(t)>d_1'(t) \,] = 1$$

According to the EDEL's learning algorithm we have:

$$d_1'(t)>d_2'(t) \Rightarrow s(t+1) = s(t) + 1 \quad (5)$$

From relation (5) is derived that:

$$\Pr[\, s(t+1)=s(t)+1] \geq \Pr[\, d_1'(t)>d_2'(t) \,] \quad (6)$$

We can select a large enough window size W, such that equality (2) gives:

$$\Pr[\, d_1'(t) > d_2'(t) \,] > 0.5 \quad (7)$$

If unequality (7) is maintained, then from unequality (6) is derived that:

$$\Pr[\, s(t+1)=s(t)+1 \,] > 0.5 \quad (8)$$

Consequently,

$$\Pr[\, s(t+1)=s(t)-1 \,] < 0.5 \quad (9)$$

Thus,

$$\Pr[s(t+1)=s(t)+1] > \Pr[s(t+1)=s(t)-1] \quad (10)$$

We note that unequalities (8),(9) and (10) are maintained independently from the specific value of s(t) (s(t)=0,1,..,N-1).

Let's define:

$$r_k = \Pr[\, s(t+1)=k+1 \mid s(t)=k \,]$$

and

$$l_k = \Pr[\, s(t+1)=k-1 \mid s(t)=k \,]$$

From (7) is derived that:

$$r_k>0.5 \quad ,k=0,1,..,N-1 \quad (11)$$

From (8) is derived that:

$$l_k<0.5 \quad ,k=1,2,..,N \quad (12)$$

Let $\pi_k=\Pr[\, s(t)=k \mid t\to\infty \,]$ the k^{th} component of the asymptotic probability vector π. It is known (reference [9]: equation 4.3.16) that under these conditions we have:

$$\pi_k = \frac{r_{k-1}}{l_k}\,\pi_{k-1}$$

Let's define:

$$q_k = \frac{r_{k-1}}{l_k}$$

From (11) and (12) is derived that:

$$q_k>1 \quad \text{for all } k=1,2,..,N \quad (13)$$

We have:

$$\lim_{t\to\infty} E[P_i(t)] = \sum_{k=0}^{N}\frac{i}{N}\,\pi_k$$

From (13) is derived that:

$$\pi_k = q_k\,\pi_{k-1} \quad \text{where } q_k>1 \text{ for } k=1,2,..,N.$$

This implies that as $N\to\infty$ the major part of the probability measure on the asymptotic probability vector π is contained in an arbitrary small neighborhood of unity.

Thus,

$$\lim_{N\to\infty}\lim_{t\to\infty} E[P_i(t)] = \lim_{N\to\infty}\sum_{k=0}^{N}\frac{i}{N}\,\pi_k \to 1 \qquad \text{q.e.d}$$

We note that the minimum window size W_{min} that guarantees the ε-optimality of the EDEL learning automaton is usually a small integer.

For example, if d_i=0.5 and d_j=0.3 then W_{min}= 3; if d_i=0.8 and d_j=0.3 then W_{min}= 1.

VII. SIMULATION RESULTS

Extended simulation results are presented that prove the EDEL's superiority compared with the classic DL$_{RP}$ ergodic scheme (proposed by Oommen in [1] and [2]) when it operates in a non-stationary environment. Both of the automata were simulated operating in Markovian switching environments. The automaton was made to switch between two environments E_1 and E_2 according to a Markov chain which determined the probability with which it was in either environment. Given that the automaton is in the E_i environment at time instant t, the probability of remaining in the same environment at time instant t+1 is equal to 1-δ (where δ is a parameter that characterizes the Markovian chain). The probability of switching to the other environment is equal to δ. More formally, we can say that if the probability of being in the E_i environment at time instant t is $P_{E_i}(t)$ then the probability of being in the E_i environment at the next time instant t+1 is:

$$P_{E_i}(t+1) = (1-\delta)\ P_{E_i}(t) + \delta\ P_{E_j}(t)$$

where $i,j \in \{1,2\}$ and $j \neq i$.

Obviously, a large δ leads to a rapidly varying Markov chain. In this case the probability of environmental switching after each step is high. A small δ leads to a slowly varying Markov chain, thus the probability of environmental switching after each step is low.

In our simulation both of the environments E_1 and E_2 offer two actions. If d_1^1 and d_2^1 are the reward probabilities of actions a_1 and a_2 correspondingly, in the E_1 environment and d_1^2 and d_2^2 are the reward probabilities of actions a_1 and a_2 correspondingly in the E_2 environment then we selected $d_1^1 = d_2^2$ and $d_1^2 = d_2^1$. Thus, when the environment switches then the reward probabilities of the two actions are interchanged.

A reliable performance index for an automaton that operates in a nonstationary environment is the average reward received by the automaton during its operation.

The average reward received R^* is computed as:

$$R^* = \frac{1}{k} \sum_{t=1}^{k} E[\ R(t)\ ]$$

where E[R(t)] is the average reward received at time instant t and k is the number of iterations done per run (k is a very large integer number). We subtract the initial average reward $R_0 = \frac{d_1+d_2}{2}$ from R^* in order to compute the automaton's power P.

Thus we have: $P = R^* - R_0$

Both automata were simulated in Markovian nonstationary environments of the type described above for various values of the δ parameter and the actions' reward probabilities.

Note, that for W=1 and N=2 the EDEL learning automaton behaves exactly as the DL$_{RP}$ automaton does for N=2. For very rapidly switching environments the choice W=1 , N=2 is the optimal one. Thus, for very rapdily switching environments (e.g. δ=0.50) the behavior of the two ergodic schemes (EDEL, DL$_{RP}$) is identical. Although, we have to note that such a value of δ (0.50) is too large. The utilization of a learning automaton in an environment that switches, on average, every two iterations has small practical interest, because the automaton's (every automaton's) power $P = R^* - R_0$ is very low (closed to 0).

The comparative results for EDEL and DL$_{RP}$ learning automata are presented in figure 1.

The highest power $P = R^* - R_0$ that the EDEL and DL$_{RP}$ learning automata achieve (utilizing the optimum values of their internal parameters: N for the DL$_{RP}$ and N,W for the EDEL) operating in various Markovian environments are appeared in this figure. These results assure the theoretical comments made about the compared schemes in Section IV. In low reward probability environments the EDEL learning automaton achieves a very high performance while, in such an environment, the DL$_{RP}$'s performance is very low. For example, when the EDEL operates in a slowly switching environment (δ=0.01) that offers reward probabilities equal

REWARD PROB.	δ = 0.01		δ = 0.10	
	EDEL $R^*- R_0$	DL_{RP} $R^*- R_0$	EDEL $R^*- R_0$	DL_{RP} $R^*- R_0$
0.4 0.1	0.1054 N=4 W=6	0.0625 N=10	0.0454 N=2 W=3	0.0315 N=2
0.6 0.3	0.0973 N=2 W=4	0.0833 N=10	0.0420 N=2 W=2	0.0373 N=2
0.8 0.5	0.0984 N=2 W=2	0.0963 N=6	0.0459 N=2 W=1	0.0459 N=2
0.2 0.1	0.0165 N=14 W=5	0.0060 N=10	0.0059 N=2 W=4	0.0034 N=2
0.4 0.3	0.0140 N=2 W=5	0.0098 N=14	0.0051 N=2 W=3	0.0040 N=4
0.8 0.7	0.0151 N=2 W=2	0.0148 N=6	0.0060 N=2 W=1	0.0060 N=2
0.8 0.2	0.2616 N=2 W=3	0.2398 N=6	0.1466 N=2 W=1	0.1466 N=2

Fig.1 EDEL versus DL_{RP} in nonstationary stochastic environments.

to 0.4 and 0.1, it achieves a power ($P=R^*- R_0$) equal to **0.1054**. The DL_{RP} learning automaton achieves a power of only **0.0625** when it operates in the same environment. When the EDEL operates in a rapidly switching environment (δ=0.10) that offers reward probabilities equal to 0.2 and 0.1, it achieves a power equal to **0.0059**. The DL_{RP} learning automaton achieves a power of only **0.0034** when it operates in the same environment.

In high reward probability environments the EDEL learning automaton achieves a high performance, quite better than the DL_{RP}'s one. For example, when the EDEL operates in a slowly switching environment (δ=0.01) that offers reward probabilities equal to 0.8 and 0.5, it achieves a power equal to **0.0984**. The DL_{RP} learning achieves a power of **0.0963** when it operates in the same environment. We can perceive that the EDEL scheme achieves a high performance in both high and low reward probability environments while the DL_{RP}'s performance is very low when it operates in a low reward probability environment.

Except of the numerical results referred above, graphs that represent the EDEL's and

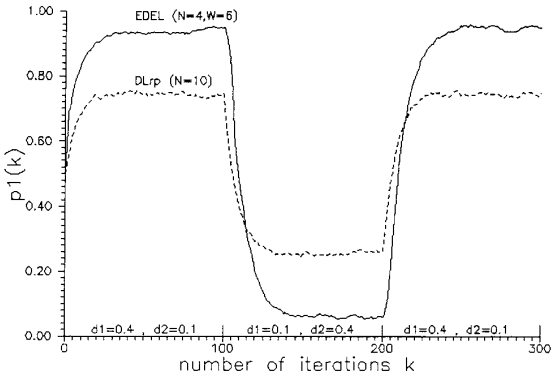

Fig.2 Graphical representation of the EDEL's and DL_{RP}'s performance in a slowly switching 0.4/0.1 environment.

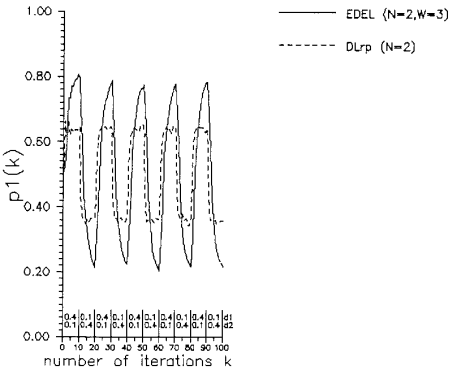

Fig.3 Graphical representation of the EDEL's and DL_{RP}'s performance in a rapidly switching 0.4/0.1 environment.

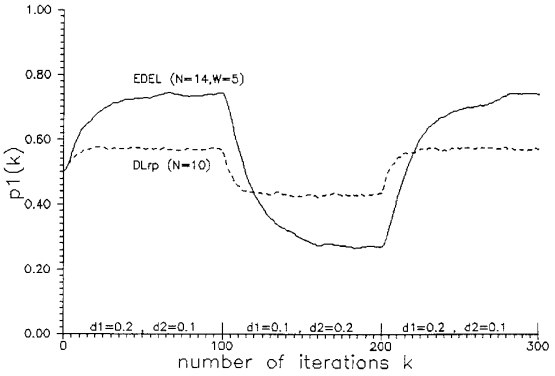

Fig.4 Graphical representation of the EDEL's and DL_{RP}'s performance in a slowly switching 0.2/0.1 environment.

DL_{RP}'s performance in a switching environment are also presented (see figures 2 to 4). These graphs represent the probability of selecting action a_1 as a function of time. The environmental changes are appeared on the graph, so we can perceive the adaptivity of both schemes to these changes. Two main results can be derived from the presented graphs:

1) The EDEL learning automaton achieves a high (closed to unity) choice probability of the optimal action (accuracy).

2) Although the above result could reduce the automaton's adaptivity to environmental changes, the EDEL scheme remains very sensitive to these changes.

VIII. CONCLUSION

The unacceptably low performance of the classic reward-penalty ergodic learning automata when they operate in high penalty emvironments has been a limiting factor in their applications. In this paper we presented a new ergodic discretized learning automaton which utilizes an estimator learning algorithm in order to achieve a high performance in every nonstationary stochastic environment.

Via extensive simulation results, we proved that the proposed EDEL learning automaton achieves a higher performance than the classic DL_{RP} ergodic scheme when they both operate in nonstationary environments.

Furthermore, we proved that the proposed EDEL learning automaton is epsilon-optimal in every stochastic environment.

We are currently working on the application of the proposed scheme in load balancing problems of computer networks.

IX. REFERENCES

[1] B.J.Oommen, "Absorbing and Ergodic Discretized two-action Learning Automata", IEEE Trans. on Syst. Man and Cybern., vol SMC-16,no.2, pp.282-293, March/April 1986.

[2] B.J.Oommen and J.P.R.Christensen, "Epsilon-optimal Discretized Linear Reward-Penalty Learning Automata",IEEE Trans. on Syst . Man and Cybern.,vol SMC-18,no.3,pp.451-458,May/June 1988.

[3] A.V.Vasilakos, G.I. Papadimitriou, C. T.Paximadis, "New Absorbing Hierarchical Discretized Pursuit Nonlinear Learning Automata with rapid convergence and high accuracy", submitted for publication.

[4] A.V.Vasilakos, V.G.Polimenis, V.E. Avgerinos "A New Nonlinear Discretized Learning Automaton with Rapid Convergence and High Accuracy", IEEE International Conference on Systems, Man andCybernetics, Cambridge, MA, Nov.1989.

[5] B.J.Oommen and J.K.Lanctot,"Epsilon-Optimal Discretized Pursuit Learning Automata", IEEE International Conference on Systems, Man and Cybernetics, Cambridge, MA, Nov.1989.

[6] K.S.Narendra and M.A.L.Thathachar, "Learning Automata-A Survey", IEEE Trans. on Syst., Man and Cybern., vol SMC-4, no.4, pp.323-334, July 1974.

[7] K.S.Narendra and S.Lakshmivarahan, "Learning Automata: A Critique", Jounal of Cybernetics and Information Sciences, Vol.1,pp.53-66,1977.

[8] M.A.L.Thathachar and P.S.Sastry, "A Class of Rapidly Converging Algorithms for Learning Automata",IEEE Trans. on Syst.Man and Cybern.,vol SMC-15, no.1, pp.168-175, January/February 1985.

[9] A.O.Allen, "Probability, Statistics and Queueing Theory with Computer Science Applications",N.York, Academic Press,1978.

[10] M.L.Tsetlin, "On the Behavior of Finite Automata in Random Media", Automat. Telemekh., vol.22, pp.1345-1354, Oct.1961.

[11] M.L.Tsetlin, "Automaton Theory and the Modeling of Biological Systems", NY, Academic, 1973.

[12] Y.Z.Tsypkin and A.S.Poznyak, "Finite Learning Automata", Engineering Cybernetics, vol.10, pp.478-490, 1972.

[13] A.V.Vasilakos,C.A.Moschonas, C.T.Paximadis, "Adaptive window flow control and learning algorithms for adaptive routing in data networks",ACM SIGMETRICS'90, 22-25 May 1990, Boulder, Colorado.

[14] A.V.Vasilakos,C.A.Moschonas, C.T.Paximadis, "Variable window flow control and ergodic discretized learning algorithms for adaptive routing in data networks", Computer Networks and ISDN Systems (to appear).