A New Approach to the Design of Reinforcement Schemes for Learning Automata: Stochastic Estimator Learning Algorithms

Georgios I. Papadimitriou

Abstract— **In this paper, a new class of learning automata is introduced. The new automata use a stochastic estimator and are able to operate in nonstationary environments with high accuracy and a high adaptation rate. According to the stochastic estimator scheme, the estimates of the mean rewards of actions are computed stochastically. So, they are not strictly dependent on the environmental responses. The dependence between the stochastic estimates and the deterministic estimator's contents is more relaxed when the latter are old and probably invalid. In this way, actions that have not been selected recently have the opportunity to be estimated as "optimal," to increase their choice probability, and, consequently, to be selected. Thus, the estimator is always recently updated and consequently is able to be adapted to environmental changes. The performance of the Stochastic Estimator Learning Automaton (SELA) is superior to the previous well-known S-model ergodic schemes. Furthermore, it is proved that SELA is absolutely expedient in every stationary S-model random environment.**

Index Terms—**Stochastic estimator, learning automaton, nonstationary random environment, adaptation rate, absolute expediency**

I. Introduction

A Learning Automaton is a finite state machine that interacts with a stochastic environment and tries to learn the optimal action offered by the environment via a learning process. The learning process is as follows (Fig. 1). The automaton chooses one of the offered actions according to a probability vector, which at every time instant contains the probability of choosing each action. The chosen action triggers the environment that responds with a feedback, depending on the mean reward of the chosen action. The automaton takes into account this answer and modifies the probability vector by means of a **learning algorithm**. A learning automaton is one that learns the action that has the maximum mean reward and that ultimately chooses this action more frequently than other actions.

According to the nature of its input, a learning automaton can be characterized as a P, Q or S-model one. If the input set is binary ($\{0, 1\}$), it is called a P-model automaton [3], [6], [7]. A learning automaton is a Q-model automaton [6], [7] if the input set is a finite set of distinct symbols. Finally, if the automaton's input can take any real value in the [0, 1] range, the automaton is called an S-model one [4], [5], [6], [7].

With respect to their Markovian representation, Learning Automata are classified into two main categories: ergodic [6], [7] or automata possessing absorbing barriers [6], [7]. The ergodic automata converge with a distribution independent of the initial state. If the stochastic characteristics of the actions are not stable (nonstationary environment), ergodic automata are preferred. Important results of the application of learning automata in computer networks applications can be found in [8] and [9].

Manuscript received March 18, 1991; revised October 2, 1992.

The author is with the Department of Computer Engineering, University of Patras, 26500 Patras, Greece, and the Computer Technology Institute, 26110 Patras, Greece.

IEEE Log Number 9212686.

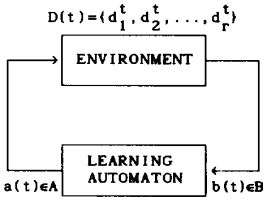

Fig. 1. A learning automaton that interacts with a random environment.

The slow convergence rate of learning automata has been a limiting factor in their applications. Estimator learning algorithms were introduced by Thathachar and Sastry [2] as an effort to solve this problem by using running estimates of the environmental characteristics.

All classic 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 environment's reward) at the next time instant. Otherwise, the choice probability of the selected action is decreased. Estimator learning algorithms are characterized by the use of a running estimate of the mean reward of each action. The change of the probability of choosing an action is based on its current estimated mean reward, rather than on the feedback of the environment. The environment determines the probability vector, not directly, but indirectly, by determining the estimates of the actions' mean rewards. 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 mean reward. Simulation results have shown the superiority of the estimator learning algorithms over the classic learning algorithms [2], [3].

The performance of estimator learning algorithms decreases when they operate in a nonstationary stochastic environment. This is due to the existence of old, and consequently invalid, feedback information in the estimator.

In this paper, we present a **stochastic estimator** scheme that in combination with the window technique presented in [2], gives a satisfactory solution to the low adaptation rate problem of S-model learning automata.

According to this scheme, the estimates of the mean rewards of actions are computed stochastically. So, they are not strictly dependent on the environmental responses. The dependence between the stochastic estimates and the deterministic estimator's contents is more relaxed when the latter are old and probably invalid. In this way, actions that have not been selected recently, have the opportunity to be estimated as "optimal," to increase their choice probability, and, consequently, to be selected. Thus, the estimator is always recently updated, and, consequently, is able to be adapted to environmental changes. Extensive simulation results indicate that the proposed stochastic estimator learning automaton achieves a significantly higher performance than the previous well-known S-model ergodic schemes when they operate in nonstationary random environments.

The structure of this paper is as follows. Section II introduces the reader to the Stochastic Estimator innovation. The presentation of the Stochastic Estimator Learning Automaton in Section III is

followed by the proof of its absolute expediency in Section IV. Extensive simulation results that indicate the superiority of the SELA's performance are presented in Section V. Finally, a brief discussion of the proposed scheme closes the paper in Section VI.

II. The Stochastic Estimator

Assume a learning automaton that operates in a nonstationary environment and keeps estimates of the actions' mean rewards. As the time passed from the last selection of an action a_i increases, there is an increase in the probability of environmental switching during this time. Consequently, the probability that a not updated estimation is still valid gradually decreases. An estimator that contains "old" and consequently invalid feedback information leads to an automaton that is incapable of being adapted to environmental changes.

Although it may look strange, the main disadvantage of a classic estimator is its absolute confidence to its contents. An estimator specially designed for operation in nonstationary environments must be able to "doubt" the validation of "old" environmental responses, and, simultaneously, give to actions that have not been selected recently the opportunity to be selected.

In this paper, we present an estimator scheme that expresses its "doubt" about the validity of its "old" contents by adding a **zero mean normally distributed** random number to each action's estimate. The **variance** of the normal distribution differs from action to action and is proportional to the time passed from the last time that each action was selected. In this way, it gives to actions that have much time to be selected the opportunity to be estimated as "optimal," to increase their choice probability, and, consequently, to be selected. This kind of estimator, which determines the estimated mean rewards of the actions in a nondeterministic way, is called a **stochastic estimator.** As simulation results show, the use of a stochastic estimator leads to a dramatic improvement of the automaton's performance when it operates in a nonstationary stochastic environment.

III. The Stochastic Estimator Learning Automaton (SELA)

The SELA learning automaton is defined by the quintuple $\langle A, B, P, E, T\rangle$ where $A = \{a_1, a_2, \cdots, a_r\}$ is the set of the r offered actions ($2 \le r < \infty$). The action selected at time instant t is denoted by $a(t)$. $B = [0, 1]$ is the input set of the possible environmental responses. The environmental response can take any value in the [0, 1] interval. The environmental response at time instant t is denoted by $b(t)$. $\mathbf{P}$ is a probability distribution over the set of actions. We have $\mathbf{P}(t) = \{p_1(t), p_2(t), \cdots, p_r(t)\}$, where $p_i(t)$ is the probability of selecting action a_i at time instant t. E is the **estimator** that, at any time instant, contains the estimated environmental characteristics. We define $E(t) = (D'(t), M(t), U(t))$ where $D'(t) = \{d'_1(t), d'_2(t), \cdots, d'_r(t)\}$ is the **Deterministic Estimator Vector**, which, at any time instant t, contains the current deterministic estimates of the mean rewards of the actions. The current deterministic estimate $d'_i(t)$ of the mean reward of action a_i is defined as follows:

$$d'_i(t) = \frac{\left(\begin{array}{c}\text{The total reward received by the automaton during}\\ \text{the last } \mathbf{W} \text{ times that action } a_i \text{ was selected.}\end{array}\right)}{\mathbf{W}} = \frac{\sum_{k=1}^{W} w_i^k(t)}{\mathbf{W}}, \quad (1)$$

where $\mathbf{W}$ is an integer internal automaton's parameter called "learning window" and $w_i^k(t)$ for $k = 1, 2, \cdots, \mathbf{W}$ are the rewards received at each one of the W last times that action a_i was selected. $\mathbf{M}(t) = \{m_1(t), m_2(t), \cdots, m_r(t)\}$ is the **Oldness Vector**, which, at any time instant, contains the time passed (time is counted in number of iterations) from the last time each action was selected. Thus, for every action a_i, we define $m_i(t) = t - \max_T\{T : T \le t \text{ and } a(T) = a_i\}$. $U(t) = \{u_1(t), u_2(t), \cdots, u_r(t)\}$ is the **Stochastic Estimator Vector**, which, at any time instant t, contains the current **stochastic** estimates of the mean rewards of the actions. The current stochastic estimate $u_i(t)$ of the mean reward of action a_i is defined as follows:

$$u_i(t) = d'_i(t) + N(0, s_i^2(t)) \quad \text{where } s_i(t) = \min\{\alpha m_i(t), s_{\max}\} \quad (2)$$

$N(0, s_i^2(t))$ symbolizes a **random number** selected with a **normal** probability distribution, with a mean equal to 0 and a variance equal to $s_i^2(t)$. α is an internal automaton's parameter that determines how rapidly the stochastic estimates become independent from the deterministic ones. $s_{\max}$ is the maximum permitted value of $s_i(t)$ ($i = 1, 2, \cdots, r$). It limits the variance of the stochastic estimates in order not to increase infinitely.

T is the learning algorithm. Its algorithmic description is presented below.

STEP 1: Select an action $a(t) = a_k$ according to the probability vector.

STEP 2: Receive the feedback $b(t) \in [0, 1]$ from the environment.

STEP 3: Compute the new **deterministic** estimate $d'_k(t)$ of the mean reward of action a_k as it is given by relation (1).

STEP 4: Update the **Oldness Vector** by setting $m_k(t) = 0$ and $m_i(t) = m_i(t-1) + 1$ for all $i \ne k$.

STEP 5: For every action $a_i (i = 1, 2, \cdots, r)$, compute the new **stochastic** estimate $u_i(t)$ as it is given by relation (2).

STEP 6: Select the "optimal" action a_m that has the highest **stochastic** estimate of mean reward. Thus, $u_m(t) = \max_i\{u_i(t)\}$.

STEP 7: Update the probability vector in the following way. For every action $a_i (i = 1, 2, \cdots, m-1, m+1, \cdots, r)$, with $p_i(t) \ge 1/N$, set the following condition:

$$p_i(t+1) := p_i(t) - 1/N.$$

For the "optimal" action a_m set, $p_m(t+1) := 1 - \sum_{i \ne m} p_i(t+1)$.

STEP 8: Go to step 1.

NOTES: N is an internal automaton's parameter, which is called "resolution parameter" and determines the step size $\Delta (\Delta = 1/N)$ of the probability updating. Δ is also called "probability slice." The initial probability distribution is computed as follows. $\lfloor N/r \rfloor$ probability slices Δ are equally distributed to all of the actions. After this distribution, if there are remaining probability slices, they are randomly distributed to the actions.

The environment in which the automaton operates is defined by the triple $\langle A, L, B\rangle$, where A and B are as defined above and $L(t) = (D(t), F(t))$. We define $D(t) = \{d_1^t, d_2^t, \cdots, d_r^t\}$ is the set that contains the mean rewards of the actions at any time instant t. Thus, $d_i^t = E[b(t) \mid a(t) = a_i \in A]$. $F(t) = \{f_1^t(x), f_2^t(x), \cdots, f_r^t(x)\}$ is the set that contains the probability density functions of the actions' rewards at every time instant t. Given action a_i is selected at time instant t, then the environment responses with a reward taken with a mean d_i^t and a density function $f_i^t(x) (-\infty < x < +\infty)$. Usually, $f_i^t(x)$ is symmetric about the line $x = d_i^t$. An environment is called a "stationary" one if the means and the density functions of the actions' rewards are time-invariant, and as a "nonstationary" one if they are time-variant.

At time instant t, the expected reward is $R(t) = \sum_{i=1}^{r} d_i^t\, p_i(t)$. Assume a stationary random environment. A learning automaton is said to be "absolutely expedient" [1], [6] if $E[R(t+1) \mid \mathbf{P}(t)] > R(t)$ for all t, all $p_k(t) \in (0, 1) (k = 1, \cdots, r)$, and all possible values of $d_i (i = 1, \cdots, r)$.

TABLE I
THE POWER OF $P = P^* - P_0$ OF SELA, $G2$ AND SLrp LEARNING AUTOMATA IN NONSTATIONARY RANDOM ENVIRONMENTS

δ	σ^2	SELA POWER	G2 POWER	SLrp POWER
0.10	0.01	0.144	0.063	0.053
	0.25	0.114	0.036	0.048
0.01	0.01	0.233	0.183	0.136
	0.25	0.230	0.152	0.134

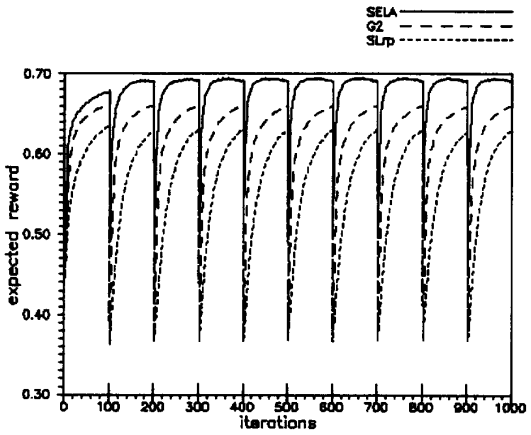

Fig. 2. Expected reward vs. iterations in a low-noise, slowly switching environment.

IV. PROOF OF ABSOLUTE EXPEDIENCY

Theorem 1: The SELA learning automaton is absolutely expedient in every stationary random environment, where the actions' rewards are symmetrically distributed about their means (thus, the environmental noise is symmetric). Thus, if $R(t) = \sum_{i=1}^{r} d_i p_i(t)$, then $E[R(t+1) \mid \mathbf{P}(t)] > \mathbf{R}(t)$ for all t, for all $p_i(t) \in (0,1)$, $i = 1, 2, \cdots, r$, and for all of the possible values of the mean rewards $d_i, i = 1, 2, \cdots, r$.

Note: The assumption of symmetrically distributed actions' rewards is not arbitrary. In all known S-model stochastic environments, the actions' rewards are symmetrically distributed about their means.

Proof: The proof is given in the Appendix.

V. SIMULATION RESULTS

The superiority of the Stochastic Estimator Learning Automaton (SELA) was affirmed in practice via extensive simulation results. The SELA was compared with other S-model ergodic schemes as the classic SLrp learning automaton [4], [6], [8] and the gradient projection–based $G2$ scheme [5].

All of the automata were simulated operating in Markovian switching environments. The automaton was made to cyclically switch between five environments E_1, E_2, E_3, E_4, and E_5 according to a Markov chain that determined the probability with which it switched from one environment to the next one. Given the fact that the automaton was in the $E_i (i \in \{1, \cdots, 5\})$ environment at time instant t, the probability of remaining in the same environment at time instant $t+1$ is equal to $1 - \delta$ (where δ is a parameter that characterizes the Markovian chain). The probability of switching to the **next environment** E_j (with $j = (i \bmod 5) + 1$) is equal to δ.

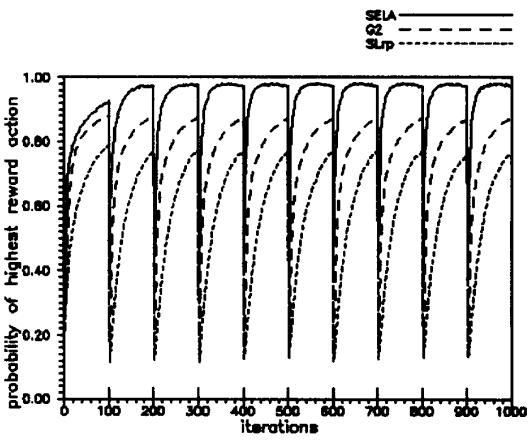

Fig. 3. Optimal action's probability vs. iterations in a low-noise, slowly switching environment.

In our simulation, all of the environments offer five actions with the following mean rewards: 0.70, 0.45, 0.40, 0.35, and 0.30. When an environment switches to the next one, then the mean rewards of the actions are cyclically shifted by one place. Thus, $E_1 = \{0.70, 0.45, 0.40, 0.35, 0.30\}$, $E_2 = \{0.30, 0.70, 0.45, 0.40, 0.35\}$, etc.

The **noise** in the environment was simulated by taking truncated samples from normal (Gaussian) distributions with the above mean values and a variance σ^2.

A reliable performance index for an automaton that operates in a nonstationary environment is the quantity of reward received by the automaton during its operation. The average expected reward R^* is computed as $R^* = \frac{1}{k} \sum_{t=1}^{k} E[R(t)]$, where $E[R(t)]$ is the average expected reward at time instant t and k is the number of iterations done per run (k is a very large integer number). If the five actions are always chosen with equal probabilities (0.2), then there is no learning. In this case, the expected reward is $R_0 = \sum_{i=1}^{5} 0.2\, d_i = 0.44$. We subtract R_0 from R^* in order to compute the automaton's **power** P. Thus, $P = R^* - R_0$. Since the optimal action has a mean reward equal to 0.70, the maximum power in such an environment is equal to $0.70 - 0.44 = \mathbf{0.26}$. The three learning automata were simulated in Markovian nonstationary environments of the type described above for various values of the δ parameter and the variance σ^2 of the Gaussian environmental noise.

The power $P = R^* - R_0$ that each automaton achieves by using the optimum values of its internal parameters ($\alpha, W, N, s_{\max}$ for SELA; $a, W, q_{\min}$ for $G2$ [5]; and a, b for SLrp [4], [6], [8]) for various values of the environmental parameters δ and σ^2 appear in Table I. These results indicate the superiority of SELA among the previous schemes in both rapidly and slowly switching environments; in both high- and low-noise environments. We can perceive that the Stochastic Estimator scheme achieves a very high power (close to the maximum one) in both high and low noise, and also in both rapidly and slowly switching environments.

Except for the numerical results to which we refer above, graphs that represent the performance of SELA, $G2$, and SLrp in switching environments are also presented (Figs. 2–9). The only difference between these environments and the ones presented above is that now the environmental switchings take place at fixed time instants. So, we can study the adaptivity of the automata to these switchings. These graphs represent the average expected reward (Figs. 2, 4, 6, 8) and the average probability of selecting the optimal action (Figs. 3,

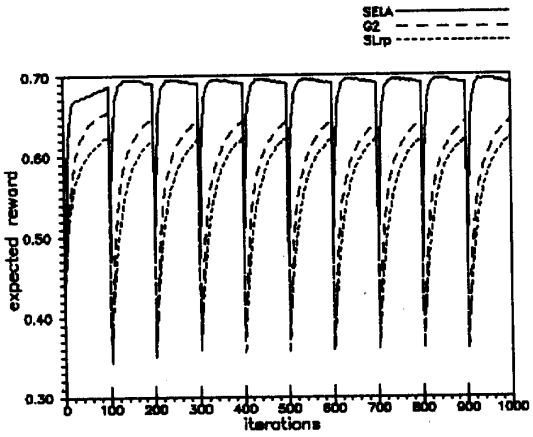

Fig. 4. Expected reward vs. iterations in a high-noise, slowly switching environment.

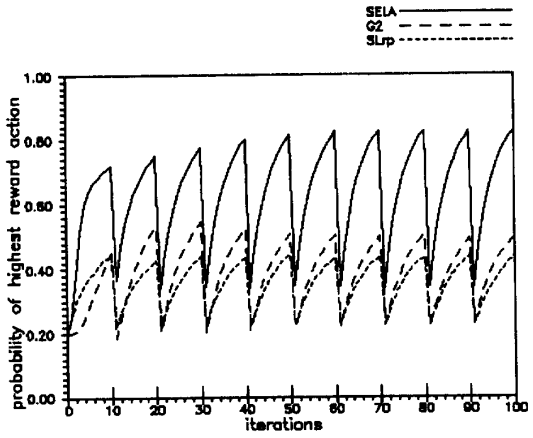

Fig. 7. Optimal action's probability vs. iterations in a low-noise, rapidly switching environment.

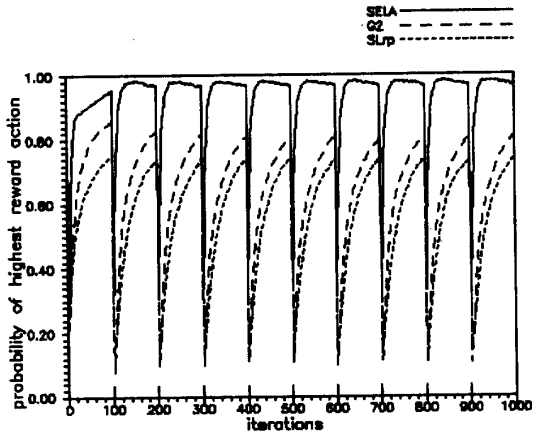

Fig. 5. Optimal action's probability vs. iterations in a high-noise, slowly switching environment.

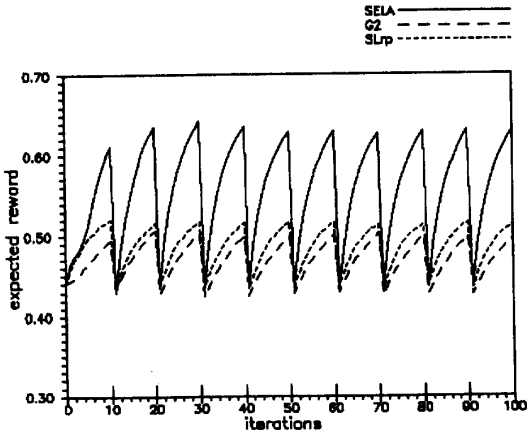

Fig. 8. Expected reward vs. iterations in a high-noise, rapidly switching environment.

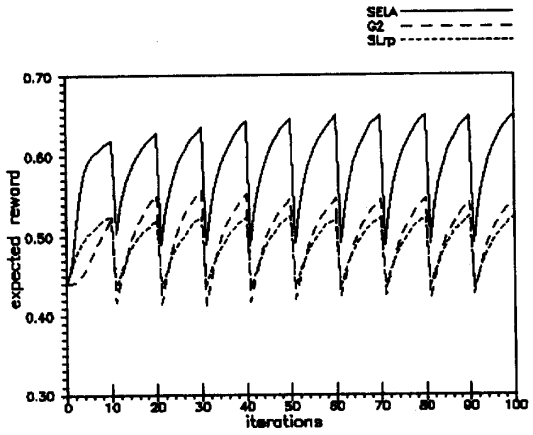

Fig. 6. Expected reward vs. iterations in a low-noise, rapidly switching environment.

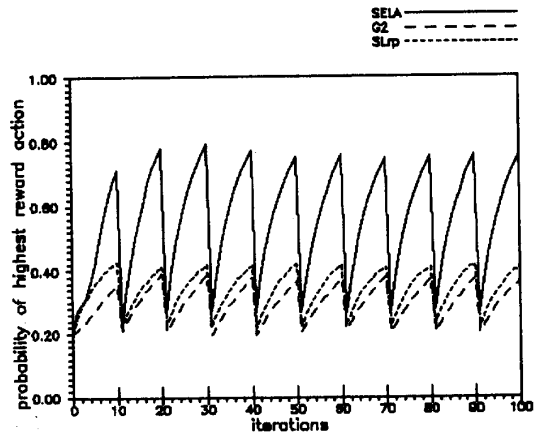

Fig. 9. Optmal action's probability vs. iterations in a high-noise, rapidly switching environment.

5, 7, 9) as a function of time. The environmental switchings appear on the iterations' axis of the graphs.

Three main results can be derived from the presented graphs.

1) The SELA learning automaton achieves a high (close to unity) choice probability of the optimal action (accuracy).
2) Although the above result could reduce the automaton's adaptivity to environmental changes, the SELA scheme remains very sensitive to these changes.
3) As a result of its high accuracy and its high rate of adaptation to environmental changes, the SELA learning automaton achieves a high power in any nonstationary stochastic environment.

Furthermore, simulation was performed in order to perceive whether the high performance of the SELA scheme is due to the use of the stochastic estimator or is due solely to the properly chosen values of W and N. The SELA scheme was simulated to operate with $\alpha = 0$ and optimal values of W and N. When $\alpha = 0$, no noise is added to the deterministic estimates. So, we can study the performance of the automaton when deterministic estimates are used. The result of this simulation is the following. For any values of the environmental parameters (δ and σ^2), the power P of the automaton takes values close to 0 when deterministic estimates are used ($\alpha = 0$). It is clear that the use of deterministic estimates leads to a dramatic decrease of the automaton's power. Therefore, we can perceive that the high power of the SELA scheme is due to the use of the stochastic estimator.

VI. Conclusion

A new S-model ergodic learning automaton that uses a **stochastic estimator** in order to achieve a high adaptation rate and a high accuracy in nonstationary random environments is introduced. Extensive simulation results are presented that indicate that the proposed SELA scheme achieves a superior performance over the previous well-known S-model ergodic schemes when they operate in nonstationary random environments. Furthermore, it is proved that the proposed SELA learning automaton is absolutely expedient in every stationary S-model random environment.

The stochastic estimator innovation can be the base of a new generation of powerful learning automata with a large number of applications.

Appendix

PROOF OF THEOREM 1: To prove Theorem 1, we first prove the following three lemmas.

Lemma 1: For any two actions a_i and a_j, let us define $P_j^i(t) = Pr[u_i(t) > u_j(t)]$. If $d_i > d_j$, then $P_j^i(t) > P_i^j(t)$ for any time instant t.

Proof: If action a_i was at last selected at time $t - m_i(t)$, and $\alpha\ m_i(t) \le s_{\max}$, then the stochastic estimate $u_i(t)$ is defined as follows:

$$u_i(t) = \frac{\sum_{k=1}^{\mathbf{W}} w_i^k(t)}{\mathbf{W}} + V \qquad \text{where } V = N(0, \alpha^2 m_i^2(t)).$$

Since $w_i^k(t) (k = 1, \cdots, \mathbf{W})$ and V are random variables symmetrically distributed about their means, it follows that $u_i(t)$ is also symmetrically distributed about its mean. Its mean $E[u_i(t)]$ and its variance $q_i^2(t)$ are as follows (where d_i and σ_i^2 denote the mean value and the variance of the reward of action $a_i (i = 1, \cdots, r)$, correspondingly):

$$E[u_i(t)] = \frac{\mathbf{W} d_i}{\mathbf{W}} + 0 = d_i \text{ and } \operatorname{Var}[u_i(t)] = W\sigma_i^2 + \alpha^2 m_i^2(t).$$

In the same way, if $\alpha\ u_i(t) > s_{\max}$, then $E[u_i(t)] = d_i$ and $\operatorname{Var}[u_i(t)] = W\sigma_i^2 + s_{\max}^2$. Let us define the random variable $Z_j^i(t) = u_i(t) - u_j(t)$. Let $f(x)$ be the density function of $Z_j^i(t)$. As discussed earlier, $f(x)$ is symmetric about the line $x = d_i - d_j$. If we define $P_j^i(t) = \Pr[u_i(t) > u_j(t)]$, we have $Z_j^i(t) > 0 \Leftrightarrow u_i(t) > u_j(t)$. Thus, we have $P_j^i(t) = \int_0^{+\infty} f(x)dx$. It is known that $d_i - d_j > 0$ and $f(x)$ is symmetric about the line $x = d_i - d_j$. Therefore, we have the following equation:

$$P_j^i(t) = \int_0^{d_i - d_j} f(x)\,dx + \int_{d_i - d_j}^{+\infty} f(x)\,dx = \int_0^{d_i - d_j} f(x)\,dx + 0.5 = Q_{ij}^t + 0.5 \Rightarrow P_j^i(t) = 0.5 + Q_{ij}^t$$

where $Q_{ij}^t = \int_0^{d_i - d_j} f(x)\,dx$. Since the variance of $Z_j^i(t)$ is bounded ($\operatorname{Var}[Z_j^i] \le \mathbf{W}\sigma_i^2 + \mathbf{W}\sigma_j^2 + 2s_{\max}^2$), it is derived that $Q_{ij}^t > 0$ for all t. Thus, $P_j^i(t) > 0.5$. Because $P_j^i(t) + P_i^j(t) = 1$ and $P_j^i(t) > 0.5$, it follows that $P_j^i(t) > P_i^j(t)$ for all t.

Lemma 2: Assume any subset A_k of the action set A, such that $|A_k| = k \le r$. Thus, $A_k = \{a_{i_1}, a_{i_2}, \cdots, a_{i_k}\} \subseteq A = \{a_1, a_2, \cdots, a_r\}$. For any action $a_i \in A_k$, let us define $B_i(A_k, t) = \Pr[u_i(t) = \max_j \{u_j(t)\}$ for $j = i_1, i_2, \cdots, i_k]$. Then, according to the SELA learning algorithm, for any two actions a_i, a_j such that $a_i \in A_k$ and $a_j \in A_k$, and for any time instant t, we have the following equation:

$$\frac{B_i(A_k, t)}{B_j(A_k, t)} = \frac{P_j^i(t)}{P_i^j(t)}.$$

Proof: The above lemma is proved by using mathematical induction on the size k of the A_k subset.

Lemma 3: For any two actions a_i and a_j, let us define the quantity $B_i(t) = \Pr[u_i(t) = \max_l \{u_l(t)\}$ for $L = 1, \cdots, r]$. If $d_i > d_j$, then $B_i(t) > B_j(t)$ for any t.

Proof: By using lemma 1 and lemma 2(for $k = r$), the above lemma follows in a straightforward manner.

Theorem 1: The SELA learning automaton is absolutely expedient in every stationary random environment that offers symmetrically distributed noise. Thus, if $R(t) = \sum_{i=1}^r d_i p_i(t)$, then $E[R(t+1)|\mathbf{P}(t)] > R(t)$ for all t, for all $p_i(t) \in (0,1)$, $i = 1, 2, \cdots, r$, and for all possible values of $d_i, i = 1, 2, \cdots, r$, assuming that the maximum reward (let d_1) is unique.

Proof: Assume that at a time instant t, we have $p_i(t) \in (0,1)$ for all $i = 1, 2, \cdots, r$. Let us define $\delta p_i(t) = p_i(t+1) - p_i(t)$ for all $i = 1, 2, \cdots, r$. The mean value of $\delta p_i(t)$ is as follows:

$$E[\delta p_i(t)] = B_i(t)(r-1)(1/N) - (1 - B_i(t))(1/N) = (1/N)(rB_i(t) - 1)$$

Therefore, for any two actions a_i and a_j, we have the following equation:

$$E[\delta p_i(t)] - E[\delta p_j(t)] = (r/N)(B_i(t) - B_j(t)).$$

Since lemma 3 guarantees that if $d_i > d_j$, then $B_i(t) > B_j(t)$ and the above relation guarantees that if $B_i(t) > B_j(t)$, then $E[\delta p_i(t)] - E[\delta p_j(t)] > 0$, we have the following equation:

$$\text{if } d_i > d_j \text{ then } E[\delta p_i(t)] > E[\delta p_j(t)] \tag{3}$$

Now compute the quantity $E[R(t+1)|\mathbf{P}(t)]$. We have the following equation:

$$E[R(t+1)] = E\left[\sum_{i=1}^r d_i(p_i(t) + \delta p_i(t))\right] = R(t) + \sum_{i=1}^r d_i\ E[\delta p_i(t)].$$

In order to prove that the proposed SELA scheme is absolutely expedient, it remains to show that $\sum_{i=1}^{r} d_i\ E[\delta p_i(t)] > 0$ for all t. Obviously, $\sum_{i=1}^{r} E[\delta p_i(t)] = 0$. (4)

Assume that actions are numbered according to the value of their mean rewards. Thus, action a_1 has the highest mean reward d_1, action a_2 has the second one, etc. Since $d_1 > d_i$ for all $i = 2, \cdots, r$, from (3), it is derived that $E[\delta p_1(t)] > E[\delta p_i(t)]$ for all $i = 2, \cdots, r$ and all t. By combining the above result with relation (4), it is proved that $E[\delta p_1(t)] > 0$ for all t. Now relation (4) guarantees the following:

$$\sum_{i=k}^{r} E[\delta p_i(t)] < 0 \text{ for all } k \geq 2 \text{ and all } t. \quad (5)$$

We are now ready to prove that $\sum_{i=1}^{r} d_i\ E[\delta p_i(t)] > 0$ for all t. We have the following conditions:

$$\sum_{i=1}^{r} d_i\ E[\delta p_i(t)] = \sum_{i=1}^{r-2} d_i E[\delta p_i(t)] + d_{r-1} E[\delta p_{r-1}(t)] + d_r E[\delta p_r(t)]$$

(Since $d_{r-1} \geq d_r$ and (5) $\Rightarrow E[\delta p_r(t)] < 0$)

$$\geq \sum_{i=1}^{r-2} d_i\ E[\delta p_i(t)]\ + d_{r-1}(E[\delta p_{r-1}(t)] + E[\delta p_r(t)]) =$$

$$\sum_{i=1}^{r-3} d_i\ E[\delta p_i(t)] + d_{r-2} E[\delta p_{r-2}(t)] + d_{r-1}(\sum_{i=r-1}^{r} E[\delta p_i(t)])$$

(Since $d_{r-2} \geq d_{r-1}$ and (5) $\Rightarrow \sum_{i=r-1}^{r} E[\delta p_i(t)] < 0$)

$$\geq \sum_{i=1}^{r-3} d_i\ E[\delta p_i(t)]\ + d_{r-2}(\sum_{i=r-2}^{r} E[\delta p_i(t)])$$

.....

$$\geq d_1\ E[\delta p_1(t)]\ + d_2(\sum_{i=2}^{r} E[\delta p_i(t)]) =$$

(Since (4) $\Rightarrow \sum_{i=2}^{r} E[\delta p_1(t)] = -E[\delta p_i(t)]$)

$$= d_1\ E[\delta p_1(t)]\ - d_2 E[\delta p_1(t)] =$$

$$(d_1 - d_2) E[\delta p_1(t)] > 0$$

(Since $d_1 > d_2$ and $E[\delta p_1(t)] > 0$ for all t.)

Thus, we have proved that $\sum_{i=1}^{r} d_i\ E[\delta p_i(t)] > 0$ for all t. Consequently, $E[R(t+1) \mid \mathbf{P}(t)] = R(t) + \sum_{i=1}^{r} d_i\ E[\delta p_i(t)] > R(t)$. Thus, the SELA scheme is absolutely expedient. **Q.E.D.**

REFERENCES

[1] S. Lakshmivarahan and M. A. L. Thathachar, "Absolutely expedient learning algorithms for stochastic automata," *IEEE Trans. Syst., Man, Cybernetics*, vol. SMC-3, pp. 281–286, May 1973.

[2] M. A. L. Thathachar and P. S. Sastry, "A Class of rapidly converging algorithms for learning automata," *IEEE Trans. Syst., Man, Cybernetics*, vol. SMC-15, no. 1, pp. 168–175, Jan./Feb. 1985.

[3] G. I. Papadimitriou, "Hierarchical discretized pursuit nonlinear learning automata with rapid convergence and high accuracy," submitted for publication.

[4] R. Viswanathan and K. S. Narendra, "Stochastic automata models with applications to learning systems," *IEEE Trans. Syst., Man, Cybernetics*, vol. SMC-3, pp. 107–111, Jan. 1973.

[5] R. Simha and J. F. Kurose, "Relative reward strength algorithms for learning automata," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, pp. 388–398, Mar./Apr. 1989.

[6] K. S. Narendra and M. A. L. Thathachar, "Learning automata: A survey," *IEEE Trans. Syst., Man, Cybernetics*, vol. SMC-4, no. 4, pp. 323–334, July 1974.

[7] K. S. Narendra and S. Lakshmivarahan, "Learning automata: A critique," *J. Cybernetics and Inform. Sci.*, vol. 1, pp. 53–66, 1977.

[8] O. V. Nedzelnitski and K. S. Narendra, "Nonstationary models of leaning automata routing in data communication networks," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-17, no. 6, pp. 1004–1015, Nov./Dec. 1987.

[9] G. I. Papadimitriou and D. G. Maritsas, "WDM passive star networks: Receiver collisions avoidance algorithms using multifeedback learning automata," in *17th IEEE Conf. Local Comput. Networks*, Minneapolis, MN, USA, 13–16 Sept. 1992.