A New Class of ϵ-Optimal Learning Automata

Georgios I. Papadimitriou, *Senior Member, IEEE*, Maria Sklira, and Andreas S. Pomportsis

***Abstract*—A new class of P-model absorbing learning automata is introduced. The proposed automata are based on the use of a stochastic estimator in order to achieve a rapid and accurate convergence when operating in stationary random environments. According to the proposed stochastic estimator scheme, the estimates of the reward probabilities of actions are not strictly dependent on the environmental responses. The dependence between the stochastic estimates and the deterministic ones is more relaxed for actions that have been selected only a few times. In this way, actions that have been selected only a few times, have the opportunity to be estimated as "optimal," to increase their choice probability and consequently, to be selected. In this way, the estimates become more reliable and consequently, the automaton rapidly and accurately converges to the optimal action. The asymptotic behavior of the proposed scheme is analyzed and it is proved to be ϵ-optimal in every stationary random environment. Furthermore, extensive simulation results are presented that indicate that the proposed stochastic estimator scheme converges faster than the deterministic-estimator-based DP_{RI} and $DGPA$ schemes when operating in stationary P-model random environments.**

***Index Terms*—Absorbing learning automata, ϵ-optimality, P-model learning automata, stationary environments, stochastic estimator.**

I. Introduction

ADAPTIVE learning is one of the main fields of artificial intelligence. Learning automaton [1]–[17] is one of the most powerful tools in this research area. A broad range of learning automata applications are reported in the literature [18]–[37].

A learning automaton is an automaton that interacts with a random environment trying to learn the optimal action offered by the environment, via a learning process [8]. The learning process is as follows (Fig. 1). The automaton chooses one of the offered actions according to a probability vector which at any time instant contains the probability of choosing each action. The chosen action triggers the environment, which responds with an answer (reward or penalty), according to the reward probability 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 probability to be rewarded and that ultimately chooses this action more frequently than other actions.

See the survey papers by Narendra and Thathachar [2] and by Narendra and Lakshmivarahan [3] and the excellent books by Narendra and Thathachar [4] and Poznyak and Najim [5], [6] for a review of the various families of learning automata, their properties and applications. In [4], an introduction to learning automata is provided. Convergence issues, stationary and nonstationary environments, interconnected automata and applications of learning automata are discussed. Book [5] focus on theoretical issues and applications of learning automata. In [6], the focus is on the use of learning automata in stochastic optimization problems. After defining learning automata and stochastic optimization, the authors focus on how the learning automata can be applied to solve unconstrained and constrained optimization problems. The optimization of nonstationary functions and the use of learning automata for solving it are also discussed.

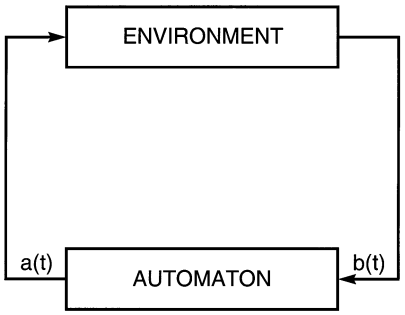


Fig. 1. Learning automaton that interacts with a random environment.

Thathachar and Sastry [16], [17] introduced the class of estimator learning automata. Estimator learning automata are characterized by the use of running estimates of the reward probabilities of actions. The change in the probability of choosing an action is based on the running estimates of the probability of reward rather than on the feedback from the environment. This means that even when an action is rewarded it is possible that the probability of choosing another action is increased. These algorithms, at every time instant, increase the probability of choosing the action with the maximum current estimate of reward probability. Simulation results have demonstrated the superiority of the estimator algorithms over the traditional learning algorithms. However, the performance of estimator learning automata is strictly dependent on the reliability of the estimator's contents. An estimator that contains unreliable data may cause a significant degradation of the accuracy and the speed of convergence.

Papadimitriou [14] introduced a new type of estimator that is called "stochastic estimator" and can be used to help the learning automaton to track rapidly switching environments. However, the stochastic estimator scheme that is presented in [14] suffers from two significant drawbacks.

a) It is not applicable to stationary environments.
b) It is not ϵ-optimal. Thus, it cannot be proved that it asymptotically converges to the optimal action.

In this paper, a new class of learning automata is introduced. The proposed automata are based on the use of a stochastic

Manuscript received May 8, 2002; revised September 3, 2002. This paper was recommended by Associate Editor B. J. Oommen.

The authors are with the Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece (e-mail: gp@csd.auth.gr).

Digital Object Identifier 10.1109/TSMCB.2003.811117

estimator in order to achieve a rapid and accurate convergence when operating in stationary environments. According to the proposed stochastic estimator scheme, the estimates of the reward probabilities of actions are computed stochastically. Therefore, they are not strictly dependent on the environmental responses. The dependence between the stochastic estimates and the deterministic ones is more relaxed for actions that have been selected only a few times. In this way, actions that have been selected only a few times have the opportunity to be estimated as "optimal," to increase their choice probability and, consequently, to be selected. In this way, the estimates become more reliable and consequently, the automaton rapidly and accurately converges to the optimal action. The asymptotic behavior of the proposed scheme is analyzed, and it is proved to be ϵ-optimal in every stationary random environment. Furthermore extensive simulation results are presented which indicate that the proposed stochastic estimator scheme achieves a faster convergence than the classic deterministic-estimator-based DP_{RI} [8]–[12] scheme and the new rapidly converging DGPA when operating in stationary P-model random environments.

The paper is organized as follows. The proposed SE_{RI} learning automaton is presented in Section II. In Section III, it is proved that the proposed scheme is ϵ-optimal. Extensive simulation results that indicate the superiority of the proposed SE_{RI} learning automaton over the well-known deterministic-estimator-based DP_{RI} and $DGPA$ schemes are presented in Section IV. Finally, concluding remarks are given in Section V.

II. SE_{RI} Learning Automaton

The stochastic estimator reward-inaction learning automaton (SE_{RI}) is a learning automaton which keeps estimates of the environmental characteristics in order to achieve a rapid and accurate convergence. The estimates of the reward probabilities 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 based on a few selections of the action. A random perturbation with zero mean and a standard deviation inversely proportional to the number of times each action has been selected is added to the deterministic estimates before selecting the action with that has the highest estimate. In this way, actions that have been selected only a few times have the opportunity to be estimated as "optimal," to increase their choice probability, and, consequently, to be selected. Thus, the estimates are more reliable and consequently, the automaton is capable of converging to the optimal action rapidly and with high accuracy.

In order to clarify the philosophy of the stochastic estimator, the following key question must be answered: *Is it possible to improve the automaton's performance by adding a perturbation that destroys the valid estimates?* The stochastic estimator selectively targets those actions that have been selected only a few times, and consequently their estimates are unreliable. Adding a random perturbation to these estimates gives them the opportunity to be estimated as "optimal," to increase their choice probability, and, consequently, to be selected. Thus, the estimates become more reliable and the automaton's performance is improved. For actions that have been selected many times, the added perturbation is very small and the deterministic estimates are practically unaffected. In other words, the stochastic estimator destroys the deterministic estimates only when they are unreliable. The key issue in SE_{RI} is the high reliability of its estimates. Based on highly reliable estimates, the SE_{RI} learning automaton rapidly converges to the optimal action.

It would be interesting to compare the proposed stochastic estimator scheme to the well-known ϵ-greedy scheme [38]. Another scheme that moves toward the direction of having reliable estimates. According to the ϵ-greedy scheme, the action with highest estimated reward probability is selected most of time, but every once in a while, say with probability x, an action is selected at random, uniformly, independently of the estimates. Although this scheme moves toward the correct direction of having reliable estimates, it suffers from two major drawbacks: i) It does not target actions which have been selected only a few times and consequently have unreliable estimates, and ii) it never selects the optimal action more than $(1-x)$ of time. A simulation based comparison of ϵ-greedy algorithms versus pursuit and linear reward-inaction algorithms is presented in [38]. The proposed SE_{RI} scheme overcomes the above drawbacks, as it is capable of targeting actions with unreliable estimates and ultimately it selects the optimal action with probability one.

The SE_{RI} learning automaton is defined as a quintuple $\langle A, B, P, E, T\rangle$ where: $A = \{a_1, \ldots, a_r\}$ is the set of the r offered actions $(2 \leq r < \infty)$. The action selected at time instant t is denoted by $a(t)$. $B = \{0, 1\}$ is the input set of the possible environmental responses. "1" denotes a reward and "0" denotes a penalty response. The environmental response at time instant t is denoted by $b(t)$. P is a probability distribution over the set of actions. We have $P(t) = \{p_1(t), \ldots, p_r(t)\}$, where $p_i(t)$ is the probability of selecting action a_i $(i = 1, \ldots, r)$ at time instant t. E is the estimator, which at any time instant t, contains the estimated environmental characteristics. We define $E(t) = (D'(t), U(t))$, where $D'(t) = \{d'_1(t), \ldots, d'_r(t)\}$ is the deterministic estimator vector, which, at any time instant t, contains the current deterministic estimates of the reward probabilities of actions. The deterministic estimate $d'_i(t)$ of the reward probability of each action a_i $(i = 1, \ldots, r)$ is defined as follows:

$$d'_i(t) = \frac{H_i^t}{G_i^t} \tag{1}$$

where G_i^t is the number of times that action a_i was selected up to time instant t, and H_i^t is the sum of the environmental responses received during these times. Thus, $H_i^t = \sum_{T:\, T \leq t \wedge a(T)=a_i} b(T)$, and $G_i^t = \sum_{T:\, T \leq t \wedge a(T)=a_i} 1$.

$U(t) = \{u_1(t), \ldots, u_r(t)\}$ is the stochastic estimator vector, which, at any time instant t, contains the current stochastic estimates of the reward probabilities of the actions. The current stochastic estimate $u_i(t)$ of each action a_i is defined as follows:

$$u_i(t) = d'_i(t) + R_i^t \tag{2}$$

where R_i^t is a random number which is uniformly distributed in the interval: $(-(\gamma/G_i^t), +(\gamma/G_i^t))$, where γ is a design pa-

rameter of the automaton which determines the magnitude of perturbation on the estimates and is called "perturbation parameter."

T is the learning algorithm. It is presented below.

STEP 0: Select each action a number of times in order to initialize H_i^0, G_i^0, and $d_i'(0)$. Set: $p_i(0) = 1/r$ $(i = 1, \ldots, r)$.

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

STEP 2: Receive the feedback $b(t) \in \{0, 1\}$ from the environment

STEP 3: Set $G_k^t := G_k^t + 1$ and $H_k^t := H_k^t + b(t)$

STEP 4: Compute the deterministic estimate $d_k'(t)$, by setting $d_k'(t) = (H_k^t/G_k^t)$

STEP 5: If $b(t) = 0$ then Goto Step 1

STEP 6: For each action a_i $(i = 1, \ldots, r)$ compute the new stochastic estimate $u_i(t)$, by setting $u_i(t) = d_i'(t) + R_i^t$

STEP 7: Select the "optimal" action a_m that has the highest stochastic estimate of reward probability. Thus, $u_m(t) = \max_i\{u_i(t)\}$.

STEP 8: Update the probability vector in the following way.

i) For every action a_i with $i \neq m$ and $p_i(t) \geq (1/(rn)))$, set $p_i(t+1) := p_i(t) - (1/(rn)))$.

(r is the number of actions, and n is the "resolution parameter" of the automaton which determines the step size of the probability updating.)

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

STEP 9: If $p_m(t+1) = 1$ then CONVERGE to action a_m else Goto step 1.

III. Proof of ϵ-Optimality

We will show that the proposed SE_{RI} learning automaton is ϵ-optimal in every stationary random environment. Thus, according to the definition of ϵ-optimality given in [11], we will show that given any (arbitrarily small) $\epsilon > 0$ and $\delta > 0$, there exists a large enough $n_0 < \infty$ (that depends on ϵ and δ) and a $t_0 < \infty$ such that for all time $t \geq t_0$ and for any resolution parameter $n > n_0$: $\Pr[|p_m(t) - 1| < \epsilon] > 1 - \delta$.

To prove the ϵ-optimality of SE_{RI} learning automaton we will use the following two theorems.

Theorem 1: Suppose that there exists an index m and a time instant $t_0 < \infty$ such that $u_m(t) > u_j(t)$ for all j with $j \neq m$ and all $t \geq t_0$. Then, there exists an integer n_0 such that for all resolution parameters $n > n_0$, $p_m \to 1$ with probability one, as $t \to \infty$.

Proof: A similar theorem was introduced and proved by Thathachar and Sastry in [16] and [17]. This theorem was extended for the case of discretized learning automata by Oommen and Lanctôt in [11]. The difference between Theorem 1 and the ones presented in [11], [16] and [17] is that the SE_{RI} scheme which is considered here is based on the use of stochastic estimates of the actions' reward probabilities rather than on deterministic ones. The proof of the present theorem follows in a straight-foreword manner from the proof of [11, Th. I] by replacing the deterministic estimates $(d_i'(t))$ with the stochastic ones $(u_i(t))$ in the formulation and the proof of the latter theorem.

Theorem 2: For action a_i, assume $p_i(0) \neq 0$. Then, for any given constants $\delta > 0$ and $M > 0$, there exists a $n_0 < \infty$ and a $t_0 < \infty$ such that for all resolution parameters $n > n_0$ and all time $t > t_0$: Pr[each action is chosen more than M times at time $t] \geq 1 - \delta$.

Proof: A similar theorem was introduced and proved in [16] and [17]. The theorem was extended for discretized learning automata in [11]. The proof of this theorem is not related to the type of the estimator (stochastic or deterministic), so the proof is identical to the one presented in [11].

Now we are ready to prove that the SE_{RI} scheme is ϵ-optimal. According to the definition of ϵ-optimality given in [11], we must prove that the following theorem holds.

Theorem 3: In every stationary random environment, the SE_{RI} is ϵ-optimal. Thus, given any $\epsilon > 0$ and $\delta > 0$, there exists a $n_0 < \infty$ (that depends on ϵ and δ) and a $t_0 < \infty$ such that for all $t \geq t_0$ and $n \geq n_0$: $\Pr[|P_m(t) - 1| < \epsilon] > 1 - \delta$.

Proof: Let ω be the difference between the two largest reward probabilities. Since the reward probability of the best action, d_m, is unique, it follows that $\omega > 0$ and $d_m - \omega \geq d_i$ for all $i \neq m$.

Let G_i^t be the number of times action a_i is chosen up to time instant t. If d_i' is the deterministic estimate of the reward probability of action a_i, then, by the weak law of large number it is derived that, for a given $\delta > 0$, there exists an $M_i^* < \infty$, such that if $G_i^t > M_i^*$:

$$\Pr\left[|d_i'(t) - d_i| < \frac{\omega}{4}\right] > 1 - \delta. \tag{3}$$

Now, let us define $M^{**} \equiv \lceil 4\gamma/\omega \rceil$. From the definition of the SE_{RI} learning algorithm, it is derived that if $G_i^t > M^{**}$, then

$$|R_i^t| < \frac{\gamma}{G_i^t} < \frac{\gamma}{4\gamma/\omega} = \frac{\omega}{4}. \tag{4}$$

Thus, for any given $\delta > 0$, there exists an $M_i \equiv \max\{M_i^*, M^{**}\}$ such that if $G_i^t > M_i$, then (3) and (4) hold.

Let us define the events

$$A \equiv |d_i'(t) - d_i| < \frac{\omega}{4}$$

and

$$B \equiv |u_i(t) - d_i| < \frac{\omega}{2}.$$

Since $|R_i^t| < (\omega/4)$, we have

$$|d_i'(t) - d_i| < \frac{\omega}{4} \Rightarrow$$

$$|d_i'(t) - d_i| + |R_i^t| < \frac{\omega}{2} \Rightarrow$$

$$\left|d_i'(t) + R_i^t - d_i\right| < \frac{\omega}{2} \Rightarrow$$

$$|u_i(t) - d_i| < \frac{\omega}{2}.$$

Thus, $A \Rightarrow B$. It is known that if $A \Rightarrow B$, then $\Pr[B] \geq \Pr[A]$. Therefore

$$\Pr\left[|u_i(t) - d_i| < \frac{\omega}{2}\right] \geq \Pr\left[|d_i'(t) - d_i| < \frac{\omega}{4}\right]. \tag{5}$$

From (3) and (5), it is derived that for any given $\delta > 0$, there exists an M_i such that if $G_i^t > M_i$

$$\Pr\left[|u_i(t) - d_i| < \frac{\omega}{2}\right] > 1 - \delta. \tag{6}$$

Let $M = \max_{1 \le i \le r} \{M_i\}$. Since ω is a positive number, it is derived that for all $j \neq m$ and for all t, if $\min_{1 \le i \le r}\{G_i^t\} > M$, then $\Pr[|d'_m(t) - d_j| < (\omega/2)]$. By Theorem 2, it is derived that we can find a t_0 and a n_0 such that for all $t > t_0$ and all $n > n_0$: $\Pr[\min_{1 \le i \le r}\{G_i^t\} > M] > 1 - \delta$.

Thus, if all actions are chosen at least M times the each of the d'_i will be in a $(\omega/2)$ neighborhood of d_i. Since $\omega > 0$, we have $d'_m(t) \ge d_m - (\omega/2) > d_i - \omega$. Therefore, for all $i \neq m$ it holds that $d'_m(t) > d'_i(t)$.

Since M is constant for each environment, by Theorem 2, it is derived that there exists a time instant $t_0 < \infty$ and a $n_0 < \infty$ such that under the SE_{RI}, for all $t > t_0$ and all $n > n_0$:

$$\Pr[G_i^t > M] \ge 1 - \delta. \tag{7}$$

From this point, the proof is the same as in [11, Th. III]. It is known that $\Pr[X] = \Pr[X|Y]\Pr[Y] + \Pr[X|Y^c](1 - \Pr[Y])$. Since probability is a continuous set function, if the following limits exist, we have

$$\lim_{t\to\infty} \Pr[X(t)] = \lim_{t\to\infty} \Pr[X(t)|Y(t)] \lim_{t\to\infty} \Pr[Y(t)] + \lim_{t\to\infty} \Pr[X(t)|Y^c(t)] \left(1 - \lim_{t\to\infty} \Pr[Y(t)]\right).$$

If we define $X(t) \equiv |p_m(t) - 1| < \epsilon$ and $Y(t) \equiv \max_i\{|d'_i(t) - d_i|\} < (\omega/2)$, then

$$\Pr[X(t)|Y(t)] = \Pr\left[|p_m(t) - 1| < \epsilon \middle| \max_i \{|d'_i(t) - d_i|\} < \frac{\omega}{2}\right].$$

From Theorems 1 and 2, it is derived that $\lim_{t\to\infty} \Pr[X(t)|Y(t)] \to 1$ because we can select a n large enough to satisfy both theorems. From Theorem 2 and relation (7), it is derived that

$$\lim_{t\to\infty} \Pr[X(t)] \ge \lim_{t\to\infty} \Pr[X(t)|Y(t)] \lim_{t\to\infty} \Pr[Y(t)].$$

Thus

$$\lim_{t\to\infty} \Pr[|p_m(t) - 1| < \epsilon] \ge 1 - \delta \qquad \text{for all } n > n_0$$

and consequently

$$\Pr[|p_m(t) - 1| < \epsilon] \ge 1 - \delta \qquad \text{for all } n > n_0 \text{ and all } t \ge t_0.$$

Q.E.D.

IV. Simulation Results

In the following, the proposed SE_{RI} learning automaton is compared to the DP_{RI} [8]–[12] and DGPA [13] schemes.

The difference between SE_{RI} and DP_{RI} lies on the type of the estimator. DP_{RI} is a deterministic-estimator-based scheme, while SE_{RI} is based on the use of a stochastic estimator. Therefore, a performance comparison between the two schemes, will clearly demonstrate the performance advantage of the stochastic estimator over the classic deterministic estimator. DGPA is a very rapidly converging learning automaton. In [13], it is shown that it converges faster than DP_{RI}. Therefore, a performance comparison between SE_{RI} and DGPA would also be useful in order to evaluate the performance of the proposed scheme.

TABLE I
ACCURACY (number of correct convergences/number of experiments) OF SE_{RI}, DP_{RI} AND DGPA IN ENVIRONMENTS E_1 TO E_5, WHEN USING THE "BEST" LEARNING PARAMETERS (250 000 EXPERIMENTS WERE PERFORMED FOR EACH SCHEME IN EACH ENVIRONMENT)

	SE_{RI}	DP_{RI}	DGPA
E_1	0.997	0.995	0.997
E_2	0.996	0.994	0.996
E_3	0.995	0.993	0.995
E_4	0.998	0.996	0.997
E_5	0.997	0.994	0.997

The three schemes which are under comparison were simulated to be applied to five stationary random environments (E_1 to E_5). The actions' reward probabilities for each environment were taken to be as follows:

a) E_1: $D = \{0.65, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10\}$.
b) E_2: $D = \{0.60, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10\}$.
c) E_3: $D = \{0.55, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10\}$.
d) E_4: $D\{$ =0.70, 0.50, 0.30, 0.20, 0.40, 0.50, 0.40, 0.30, 0.50, 0.20$\}$.
e) E_5: $D = \{0.10, 0.45, 0.84, 0.76, 0.20, 0.40, 0.60, 0.70, 0.50, 0.30\}$.

Note: Environments E_4 and E_5 are the well-known benchmark environments used in [8], [11], [13], and [16].

It has been proved that SE_{RI}, DP_{RI} and DGPA asymptotically converge to the optimal action. It remains to find out which of the three schemes converges faster. In order to compare their relative performances, we performed simulations to accurately characterize their rates of convergence. In all the tests performed an algorithm was considered to have converged if the probability of choosing an action was greater or equal to a threshold T $(0 < T \le 1)$. If the automaton converged to the action that has the highest reward probability, it was considered to have converged correctly.

Before comparing the performance of the automata, a large number of multiple tests were executed to determine the "best" values of the resolution parameter n for each scheme. We used the same method used in [8]–[10] and [13] in order to determine the "best" parameters. The values where considered as the

TABLE II
COMPARISON OF THE AVERAGE NUMBER OF ITERATIONS REQUIRED FOR CONVERGENCE OF SE_{RI}, DP_{RI} AND $DGPA$ WHEN OPERATING IN ENVIRONMENTS E_1 TO E_5. FOR ALL SCHEMES, THE "BEST" LEARNING PARAMETERS FOR EACH ENVIRONMENT ARE USED (250 000 EXPERIMENTS WERE PERFORMED FOR EACH SCHEME IN EACH ENVIRONMENT)

	SE_{RI}		DP_{RI}		DGPA	
	Parameters	Iterations	Parameters	Iterations	Parameters	Iterations
E_1 D={0.65, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10}	n=16, γ=8	426	n=298	1086	n=33	880
E_2 D={0.60, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10}	n=32, γ=12	834	n=653	2500	n=65	1677
E_3 D={0.55, 0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10}	n=105, γ=25	2540	n=2356	9613	n=204	5191
E_4 D={0.70, 0.50, 0.30, 0.20, 0.40, 0.50, 0.40, 0.30, 0.50, 0.20}	n=13, γ=6	325	n=216	783	n=28	754
E_5 D={0.10, 0.45, 0.84, 0.76, 0.20, 0.40, 0.60, 0.70, 0.50, 0.30}	n=33, γ=12	729	n=881	2363	n=55	1445

best values if they yielded the fastest convergence and the automaton converged to the correct action in a sequence of NE experiments. The values of T and NE were taken to be equal to the ones used in [8]–[10]. Thus, $T = 0.999$, and $NE = 750$.

In an effort to reduce the variance coefficient[1] of the "best" values of n that are obtained by using the above algorithm, we performed the same procedure 20 times, and we computed the average "best" value of n in these experiments. These average values were then chosen as the final "best" parameter values. In this way, the highest variance coefficient of n is significantly reduced. When 40 tests were performed, with each test consisting of one experiment, then the highest (among all environments and all automata) variance coefficient of the resulting values of n was 0.273. When 40 tests performed, with each test consisting of 20 experiments then the above variance coefficient was reduced to 0.071 which is small enough to obtain reliable results [8]. It is important to keep the variance coefficient low, because a high variance could lead to significantly different accuracies of the compared schemes and, consequently, to an unfair comparison.

As mentioned above, the SE_{RI} scheme has two learning parameters: the well-known resolution parameter n and the perturbation parameter γ. In order to evaluate the "best" values of γ and n, we repeated the above standard procedure to obtain the "best" n parameter for each value of γ. Then, we evaluated the speed of convergence for each one of the resulting pairs of n and γ by performing 5000 experiments for each pair. We considered as "best" parameter γ the one that achieved the fastest convergence when using its corresponding "best" value of n.

After determining the "best" learning parameters of each scheme for each environment, we executed 250 000 experiments for each scheme by using the "best" parameters in order to check the accuracy of each scheme when using its "best" parameters. The accuracy (*number of correct convergences/number of experiments*) of the three schemes in environments E_1 to E_5 when using their "best" parameters are presented in Table I. In each environment, the SE_{RI} achieved an accuracy equal or grater than the accuracies of DP_{RI} and DGPA. Therefore, we are sure that the performance comparison between SE_{RI} and the other two schemes is fair. The average number of iterations that SE_{RI}, DP_{RI} and $DGPA$ required for convergence to the optimal action when operating in environments E_1 to E_5 are presented in Table II. Before starting each algorithm, all actions were sampled ten times each in order to initialize the estimate vector [8]–[10]. These extra iterations are also included in the results presented in Table II.

In addition to the numerical results presented above, curves that represent the mean choice probability of the optimal action as a function of time are also presented in Figs. 2, 4, 6, 8, and 10. Finally, graphs that depict the variance coefficient of the choice probability of the optimal action as function of time are also provided in Figs. 3, 5, 7, 9, and 11. All the results presented in the above tables and graphs are averages from 250 000 experiments that were executed for each learning automaton. We checked the variance coefficient of the above results by recomputing 20 times the number of iterations and the accuracy of SE_{RI} in environment E_4. The variance coefficient was 0.0005 for the number of iterations and 0.0001 for the accuracy.

[1]The variance coefficient of a random variable X is defined as the fraction $(\sigma\{X\}/E\{X\})$, where $\sigma\{X\}$ is the standard deviation, and $E\{X\}$ is the mean value of random variable X. It expresses the standard deviation of the random variable as a percentage of its mean value.

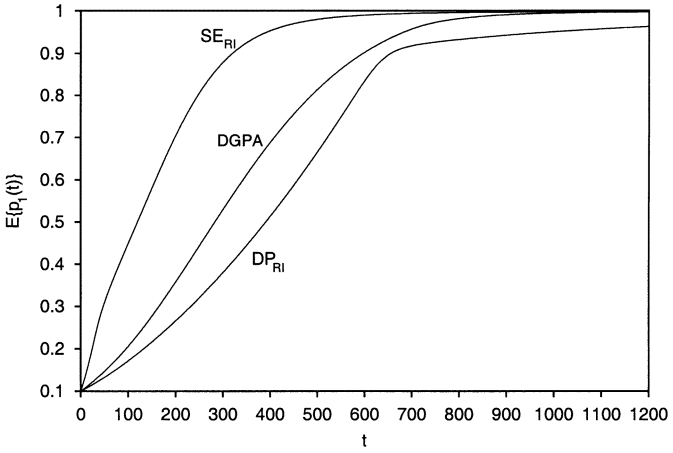


Fig. 2. $E\{p_1(t)\}$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_1. For all schemes, the "best" learning parameters are used.

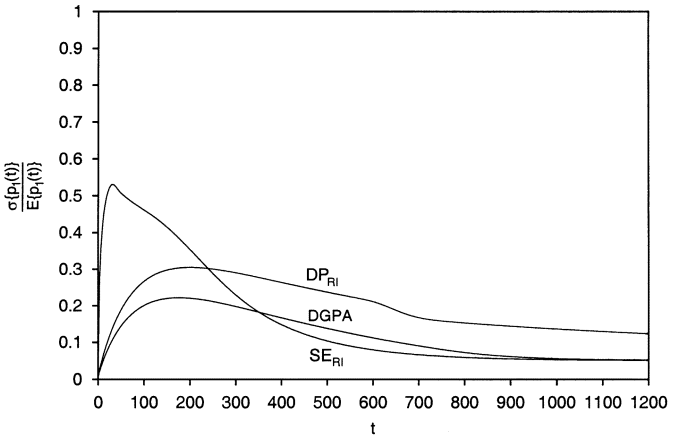


Fig. 3. Variance coefficient $(\sigma\{p_1(t)\}/E\{p_1(t)\})$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_1. For all schemes, the "best" learning parameters are used.

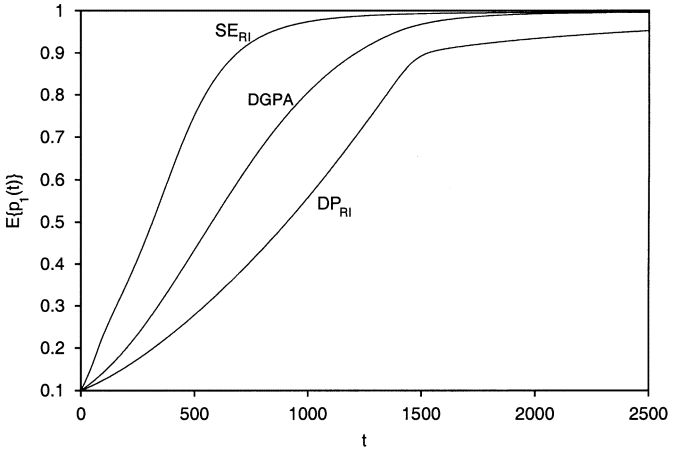


Fig. 4. $E\{p_1(t)\}$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_2. For all schemes, the "best" learning parameters are used.

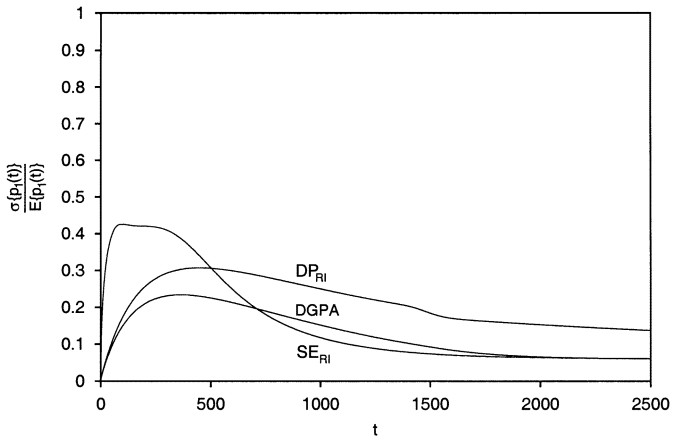


Fig. 5. Variance coefficient $(\sigma\{p_1(t)\}/E\{p_1(t)\})$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_2. For all schemes, the "best" learning parameters are used.

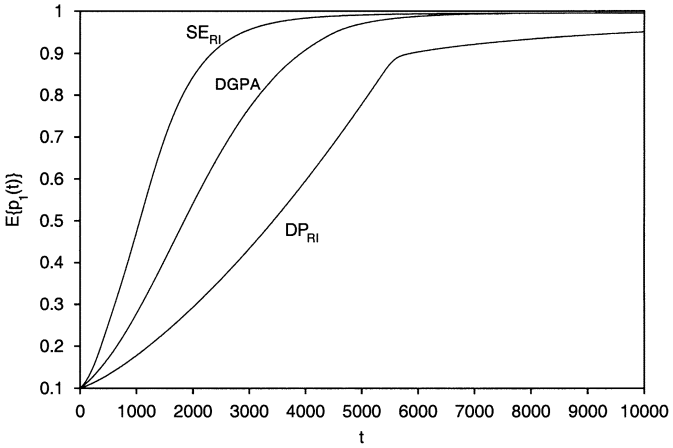


Fig. 6. $E\{p_1(t)\}$ versus t characteristics of SE_{RI}, DP_{RI}, and $DGPA$ when operating in environment E_3. For all schemes, the "best" learning parameters are used.

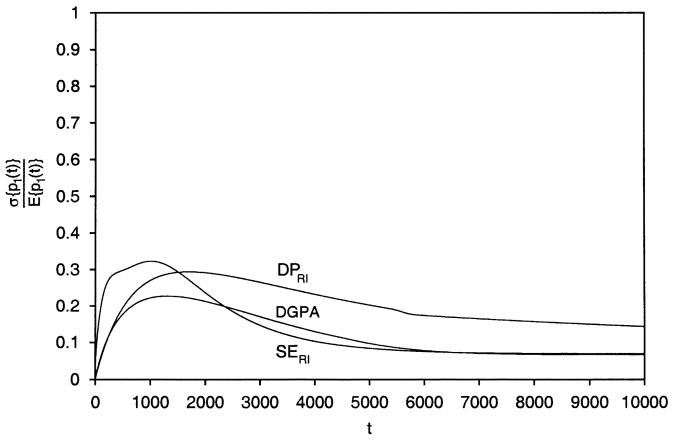


Fig. 7. Variance coefficient $(\sigma\{p_1(t)\}/E\{p_1(t)\})$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_3. For all schemes, the "best" learning parameters are used.

The following results can be extracted from the above tables and graphs.

1) The SE_{RI} scheme converges significantly faster than DP_{RI} and DGPA.

- In environment E_1 the DP_{RI} scheme requires 1086 iterations to converge, while DGPA requires 880 iterations. In the same environment, the SE_{RI} takes only 426 iterations to converge. Thus, the SE_{RI} achieves an increase in the

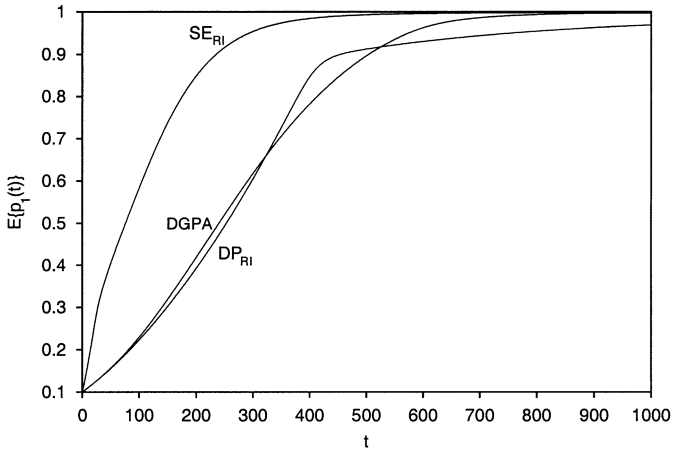


Fig. 8. $E\{p_1(t)\}$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_4. For all schemes, the "best" learning parameters are used.

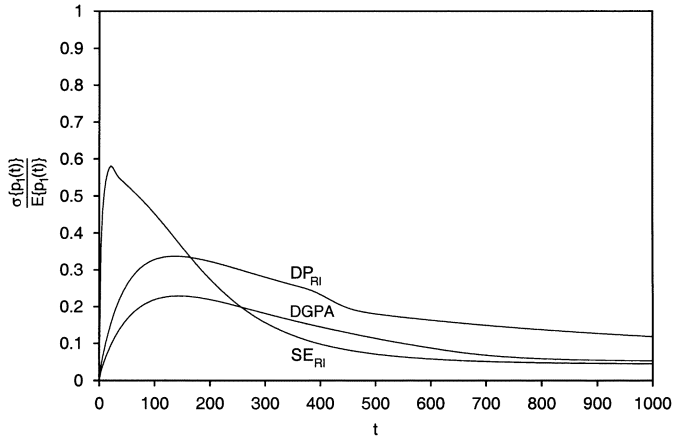


Fig. 9. Variance coefficient $(\sigma\{p_1(t)\}/E\{p_1(t)\})$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_4. For all schemes, the "best" learning parameters are used.

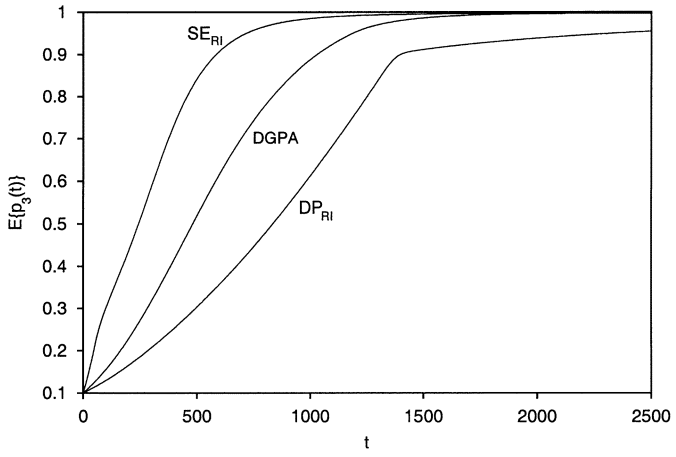


Fig. 10. $E\{p_3(t)\}$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_5. For all schemes, the "best" learning parameters are used.

speed of convergence of 61% in comparison to DP_{RI} and 52% in comparison to DGPA.

- When operating in the E_2 environment, the DP_{RI} takes 2500 iterations to converge, while the DGPA requires 1677 iterations. The SE_{RI} converges in only 834 iterations in the same environment. Thus, an improvement of 67% in comparison to DP_{RI} and 50% in comparison to DGPA is achieved.
- In the E_3 environment the SE_{RI} converges in 2540 iterations, while the DP_{RI} requires 9613 and the DGPA takes 5191 iterations to converge. Thus, the proposed SE_{RI} scheme achieves an improvement of 74% in comparison to DP_{RI} and 51% in comparison to DGPA.
- The SE_{RI} learning automaton converges in an average of 325 iterations when operating in environment E_4. In this environment the DP_{RI} converges in an average of 783 iterations, while the DGPA requires an average of 754 iterations. Thus, a performance improvement of 58% in relation to DP_{RI} and 57% in comparison to DGPA is achieved.
- Finally, when operating in environment E_5, the SE_{RI} converges in 729 iterations, while the DP_{RI} takes 2363 iterations to converge and the DGPA converges in 1445 iterations. Thus, the number of iterations required to converge is reduced by a factor of 69% in relation to DP_{RI} and 50% in comparison to DGPA.

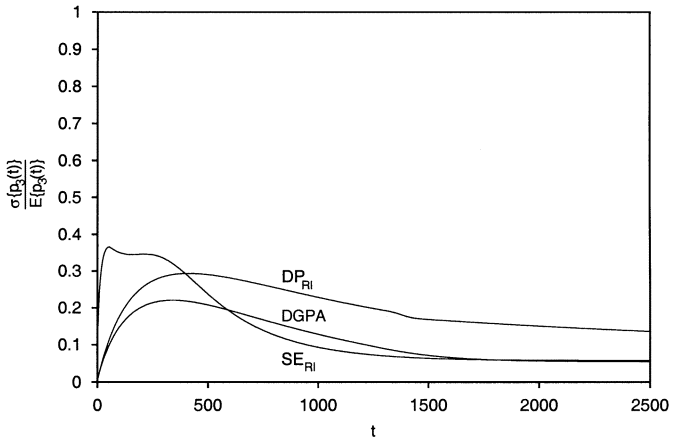


Fig. 11. Variance coefficient $(\sigma\{p_3(t)\}/E\{p_3(t)\})$ versus t characteristics of SE_{RI}, DP_{RI}, and DGPA when operating in environment E_5. For all schemes, the "best" learning parameters are used.

It is noticeable that the above reduction in the number of iterations required to converge is not achieved in the cost of reducing the accuracy of the scheme, since the SE_{RI} achieves an accuracy which is equal or greater than the ones of DP_{RI} and DGPA.

From the above results, it becomes clear that the performance advantage of SE_{RI} over DP_{RI} is stronger when the two schemes are operating in "difficult" environments (i.e., environments where the reward probability of the optimal action is close to the reward probabilities of the other actions). In "difficult" environments, the need for reliable estimates is more intense, and consequently, the use of a stochastic estimator leads to a higher performance improvement. In comparison to DGPA, the performance advantage of SE_{RI} almost the same (50%) in all environments. The probability updating scheme of DGPA is completely different, so it is difficult to say what fraction of this improvement is due to the use of the stochastic estimator.

2) For the first few iterations, the SE_{RI} scheme has a higher variance coefficient of the optimal action's choice probability than DP_{RI} and DGPA, but this situation is gradually inverted as the perturbation on the estimates of SE_{RI} is gradually decreased and the automaton converges to the optimal action.

In general, the use of the stochastic estimator leads to a decrease in the number of iterations required for convergence of about 70% in comparison to DP_{RI} and 50% in comparison to DGPA. This, is achieved in the cost of having a higher variance coefficient of the optimal action's probability during the first iterations.

V. Conclusion

A new P-model absorbing learning automaton is introduced. The proposed SE_{RI} scheme is based on the use of a stochastic estimator in order to achieve a rapid convergence. The asymptotic behavior of the proposed scheme is analyzed and it is proved that the SE_{RI} scheme is ϵ-optimal in every stationary random environment. Furthermore, extensive simulation results are presented that indicate that the proposed stochastic-estimator-based SE_{RI} scheme converges much faster than the deterministic-estimator-based DP_{RI} and DGPA. The SE_{RI} learning automaton is a powerful tool that can be applied to broad range of engineering problems. We are currently working on applying the SE_{RI} learning automaton to the bandwidth allocation problem of communication networks.

Acknowledgment

The authors are grateful to the associate editor and the reviewers for their useful and insightful comments, which helped us to improve our work.

References

[1] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, Eds., "Special issue on Learning automata: theory, paradigms and applications," in IEEE Trans. Syst., Man, Cybern. B, 2003, to be published.

[2] K. S. Narendra and M. A. L. Thathachar, "Learning automata—A survey," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp. 323–334, July 1974.

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

[4] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Applications*. New York: Pergamon, 1994.

[6] A. S. Poznyak and K. Najim, *Learning Automata and Stochastic Optimization*. New York: Springer, 1997.

[7] B. J. Oommen, "Absorbing and ergodic discretized two-action learning automata," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, pp. 282–293, Mar./Apr. 1986.

[8] B. J. Oommen and M. Agache, "Continuous and discretized pursuit learning schemes: Various algorithms and their comparison," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 277–287, June 2001.

[9] B. J. Oommen and M. Agache, "A comparison of continuous and discretized pursuit learning schemes," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Tokyo, Japan, Oct. 1999, pp. IV:1061–1067.

[10] M. Agache and B. J. Oommen, "Continuous and discretized generalized pursuit learning schemes," in *Proc. 4th World Multiconf. Systemics, Cybern.*, Orlando, FL, July 2000, pp. VII:270–275.

[11] B. J. Oommen and J. K. Lanctôt, "Discretized pursuit learning automata," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 931–938, July/Aug. 1990.

[12] ——, "Epsilon-optimal discretized pursuit learning automata," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 1, pp. 6–12.

[13] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: New families of continuous and discretized learning automata," IEEE Trans. Syst., Man, Cybern., to be published.

[14] G. I. Papadimitriou, "A new approach to the design of reinforcement schemes for learning automata: Stochastic estimator learning algorithms," *IEEE Trans. Knowledge Data Eng.*, vol. 6, pp. 649–654, Aug. 1994.

[15] ——, "Hierarchical discretized pursuit nonlinear learning automata with rapid convergence and high accuracy," *IEEE Trans. Knowledge Data Eng.*, vol. 6, pp. 654–659, Aug. 1994.

[16] M. A. L. Thathachar and P. S. Sastry, "A new approach to the design of reinforcement schemes for learning automata," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 168–175, Jan./Feb. 1985.

[17] ——, "Estimator algorithms for learning automata," in *Proc. Platinum Jubilee Conf. Syst. Signal Process.*. Bangalore, India: Dept. Elec. Eng., Indian Inst. Sci., Dec. 1986.

[18] G. I. Papadimitriou and D. G. Maritsas, "Learning automata-based receiver conflict avoidance algorithms for WDM broadcast-and-select star networks," *IEEE/ACM Trans. Networking*, vol. 4, pp. 407–412, June 1996.

[19] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Using learning automata for adaptive push-based data broadcasting in asymmetric wireless environments," *IEEE Trans. Veh. Technol.*, vol. 51, pp. 1652–1660, Nov. 2002.

[20] G. I. Papadimitriou and A. S. Pomportsis, "Self-adaptive TDMA protocols for WDM star networks: A learning-automata-based approach," *IEEE Photon. Technol. Lett.*, vol. 11, pp. 1322–1324, Oct. 1999.

[21] K. S. Narendra and M. A. L. Thathachar, "On the behavior of a learning automaton in a changing environment with application to telephone traffic routing," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-10, pp. 262–269, May 1980.

[22] B. J. Oommen and T. De St. Croix, "String taxonomy using learning automata," *IEEE Trans. Syst., Man Cybern.*, vol. 27, pp. 354–365, Apr. 1997.

[23] S. Mukhopadhyay and M. A. L. Thathachar, "Associative learning of Boolean functions," *IEEE Trans. Syst., Man Cybern.*, vol. 19, pp. 1008–1015, Sept./Oct. 1989.

[24] B. J. Oommen and D. C. Y. Ma, "Stochastic automata solutions to the object partitioning problem," *Comput. J.*, vol. 35, pp. A105–A120, 1992.

[25] ——, "Fast object partitioning using stochastic learning automata," in *Proc. Int. Conf. Res. Development Inform. Retrieval*, New Orleans, June 1987, pp. 111–122.

[26] B. J. Oommen and T. D. Roberts, "A fast and efficient solution to the capacity assignment problem using discretized learning automata," in *Proc. Eleventh Int. Conf. Industrial Eng. Applicat. Artificial Intell. Expert Syst.*, vol. II, Benicassim, Spain, June 1998, pp. 56–65.

[27] K. Najim and G. Oppenheim, "Learning systems: Theory and applications—A survey," *Proc. Inst. Elect. Eng. Comput. Digital Techn.*, vol. 138, no. 4, pp. 183–192, 1991.

[28] K. Najim and M. Chtourou, "Multilevel learning control of an absorption column," *J. Optimal Contr. Applicat. Methods*, vol. 12, no. 3, pp. 189–195, 1991.

[29] J. Valenzuela, K. Najim, R. del Villar, and M. Bourassa, "Learning control of an autogenous grinding circuit," *Int. J. Mineral Process.*, vol. 40, pp. 45–56, 1993.

[30] T. Borgers and R. Sarin, "Native reinforcement learning with endogenous aspirations," Mimeo, Univ. Coll. London, London, U.K., Texas A&M Univ. College Station, TX, 1995.

[31] G. P. Frost, T. J. Gordon, M. N. Howell, and Q. H. Wu, "Moderated reinforcement learning of active and semi-active vehicle suspension control laws," in *Proc. Inst. Mech. Engineers, J. Syst. Contr. Eng., Part I*, vol. 210, 1996, pp. 249–257.

[32] A. S. Poznyak, K. Najim, and E. Ikonen, "Adaptive selection of the optimal order of linear regression models using learning automata," *Int. J. Syst. Sci.*, vol. 27, no. 1, pp. 97–112, 1996.

[33] E. Ikonen and K. Najim, "Use of learning automata in distributed fuzzy logic processor training," *Proc. Inst. Elect. Eng. Contr. Theory Applicat.*, vol. 144, pp. 255–262, May 1997.

[34] C. Unsal, P. Kachroo, and J. S. Bay, "Multiple stochastic learning automata for vehicle path control in an automated highway system," *IEEE Trans. Syst., Man, Cybern. A*, vol. 29, no. 1, pp. 120–128, 1999.

[35] M. N. Howell and M. C. Best, "On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata," *IFAC J. Contr. Eng. Practice*, vol. 8, pp. 147–154, Feb. 2000.

[36] M. N. Howell and T. J. Gordon, "Continuous action reinforcement learning automata and their application to adaptive digital filter design," *J. Eng. Applicat. Automat. Contr.*, no. 5, Oct. 2001.

[37] I. O. Bucak and M. A. Zohdy, "Reinforcement learning control of nonlinear multi-link system," *J. Eng. Applicat. Automat. Contr.*, no. 5, Oct. 2001.

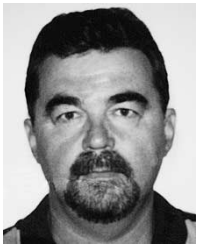
[38] R. S. Shutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

Georgios I. Papadimitriou (M'89–SM'02) received the Diploma and Ph.D. degrees in computer engineering and informatics from the University of Patras, Patras, Greece, in 1989 and 1994 respectively.

From 1989 to 1994, he was a Teaching Assistant at the Department of Computer Engineering and Informatics, University of Patras, and a Research Scientist at the Computer Technology Institute, Patras. From 1994 to 1996, he was a Postdoctorate Research Associate at the Computer Technology Institute. From 1997 to 2001, he was a Lecturer at the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece. Since 2001, has been an Assistant Professor at the Department of Informatics, Aristotle University of Thessaloniki. His research interests include design and analysis of broadband networks and learning automata. He is a member of the Editorial Board of the *International Journal of Communication Systems*. He is also an Associate Editor of the *Simulation: Transactions of the Society for Modeling and Simulation International*. He is co-author of the books *Wireless Networks* (New York: Wiley, 2002) and *Multiwavelength Optical LANs* (New York: Wiley, 2003) and is the author of more than 80 refereed journal and conference papers.

Maria Sklira received the Diploma degree in computer engineering and informatics from the University of Patras, Patras, Greece, in 1991.

Since 1992, she has been with the Informatics Division of Egnatia Bank S.A. Since 2002, she has been pursuing the Ph.D. degree at the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece. Her research interests include electronic banking and learning automata.



Andreas S. Pomportsis received the B.S. degree in physics and the M.S. degree in electronics and communications, both from the University of Thessaloniki, Thessaloniki, Greece, and a Diploma Degree in electrical engineering from the Technical University of Thessaloniki. In 1987, he received the Ph.D. degree in computer science from the University of Thessaloniki.

Currently, he is a Professor with the Department of Informatics, Aristotle University of Thessaloniki. His research interests include computer networks, learning automata, computer architecture, parallel and distributed computer systems, and multimedia systems.