Economides, A.A. & Kehagias, A.: **The STAR Automaton: expediency and optimality properties.** *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 32, No 6, pp. 723-737, Dec. 2002.

The STAR Automaton: Expediency and Optimality Properties

Anastasios A. Economides and Athanasios Kehagias

***Abstract*—We present the *STack ARchitecture* (STAR) automaton. It is a fixed structure, multiaction, reward-penalty learning automaton, characterized by a star-shaped state transition diagram. Each branch of the star contains D states associated with a particular action. The branches are connected to a central "neutral" state. The most general version of STAR involves probabilistic state transitions in response to reward and/or penalty, but deterministic transitions can also be used. The learning behavior of STAR results from the stack-like operation of the branches; the learning parameter is D. By mathematical analysis, it is shown that STAR with deterministic reward/probabilistic penalty and a sufficiently large D can be rendered ϵ-optimal in every stationary environment. By numerical simulation it is shown that in nonstationary, switching environments, STAR usually outperforms classical variable structure automata such as L_{R-P}, L_{R-I}, and $L_{R-\epsilon P}$.**

***Index Terms*—Adaptive systems, ϵ-optimality, learning automata, nonstationary environment.**

I. Introduction

EARLY work on learning developed in the context of mathematical psychology [1]–[3]. Learning is the ability to improve performance using past experience, and is necessary for adaptive decision making in a random environment with characteristics which are unknown, difficult to describe, or difficult to quantify. The theory of *learning automata* [4] provides a framework for the design of automata (i.e., simple entities) which interact with a random environment and learn dynamically the action that will produce the most desirable environment response.

At times $n = 1, 2, \ldots$, an automaton selects one of several available actions, according to action probabilities determined by the current state. The environment provides a random response to the action selected; the response can be favorable (reward) or unfavorable (penalty). Depending on the environment response, the automaton changes state. When the action probabilities of each state remain time-invariant, we have a *fixed-structure stochastic automaton (FSSA)*. When the action probabilities change in time, we have a *variable-structure stochastic automaton (VSSA)*.

The theory of learning automata was inaugurated with the study of FSSA [5]. Later, interest shifted to the study of VSSA [6] which appeared to be more adaptable. Classic examples of VSSA are L_{R-P}, L_{R-I}, and $L_{R-\epsilon P}$. An excellent overview of the theory and applications of "classical" VSSA appears in [4]. For some more recent applications, the reader is referred to [7]–[13]. New VSSA algorithms have also appeared in the literature, e.g., the so-called *estimator* algorithms [14], [15] and *pursuit* algorithms [14], [16]. An interesting development in the field of VSSA is the introduction of *discretized* VSSA. This idea has been introduced by Oommen [17]. In [18] and [19], action probabilities are updated by the usual VSSA rules; however, only a large but *finite* number of discretized probability values is used. As pointed out in [18] and [19], it is difficult to show ϵ-optimality for the multiaction discrete VSSAs. For a comparison between continuous and discretized VSSA, see [20] and [21].

It can be seen from the above references that current learning automata research is concentrated mainly on VSSAs. On the other hand, FSSs are easier to implement and require less computation per time step. This motivated us to return to the FSSA idea and search for FSSA designs which perform as well or better than corresponding VSSAs (e.g., are expedient, ϵ-optimal, converge quickly, etc.) Good performance combined with simplicity of implementation would make such FSSAs attractive competitors to the currently used VSSAs.

In this paper, we introduce the STar ARchitecture (STAR) automaton, an FSSA with the above-mentioned properties, and compare its behavior to that of several "classical" VSSAs, namely, L_{R-P}, L_{R-I}, and $L_{R-\epsilon P}$. (The comparison to discretized VSSA will be performed in a future paper.) As mentioned, we are particularly interested in the behavior of STAR in nonstationary enviroments (the importance of which is further discussed in Section II). We present computer simulations which indicate that STAR$^{(D)}$ can outperform VSSAs such as L_{R-P}, L_{R-I}, and $L_{R-\epsilon P}$. We believe that the improved performance of STAR is due to the use of a few discrete values of action probabilities. This minimizes the requirements on the random number generator and speeds up convergence.

The name "STAR automaton" refers to the star-shaped structure of the transition diagram, displayed in Fig. 1(a). Each branch of the star consists of several states, which are "committed" to one of the actions available to the automaton; in addition, each branch behaves like a stack. The *depth* D of the branches is a parameter of the automaton, hence we speak of STAR$^{(1)}$, STAR$^{(2)}$, and, in general, STAR$^{(D)}$. The depth determines speed of response and optimality. For example, we prove that in case $D = 1$, STAR$^{(1)}$ has the same limiting behavior as the L_{R-P}. In general, the D parameter can be fine-tuned to provide the best tradeoff between optimality and speed of response to switching environments.

Manuscript received November 15, 2001; revised March 1, 2002. This paper was recommended by Guest Editors M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis.

A. Economides is with the Department of Economics, University of Macedonia, 54006 Thessaloniki, Greece (e-mail: economid@macedonia.uom.gr).

A. Kehagias is with the Department of Mathematics, Physical and Computational Sciences, Faculty of Engineering, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece (e-mail: kehagias@egnatia.ee.auth.gr).

Publisher Item Identifier S 1083-4419(02)06465-8.

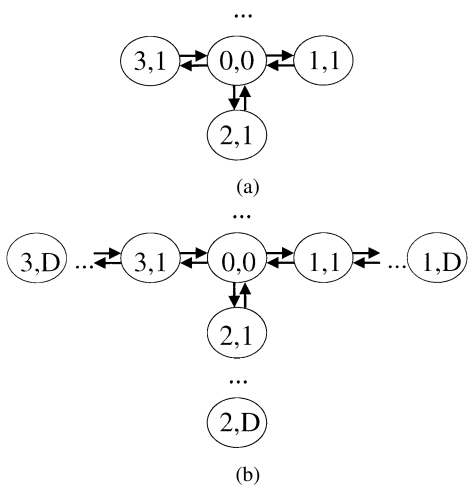


Fig. 1. (a) Structure of STAR$^{(1)}$. (b) Structure of STAR$^{(D)}$.

An essential feature of STAR is that the reward and/or penalty mechanisms can be probabilistic (depending on parameters ϵ and δ, respectively). It must be noted that *the deterministic reward/probabilistic penalty STAR*$^{(D)}$ *can become* ϵ*-optimal in any environment by appropriate choice of* D *and* δ.[1] Furthermore, numerical experiments indicate that the value $D = 2$ gives consistently good results in a wide variety of environments.

The rest of the paper is organized as follows. In Section II, we review the fundamental concepts of stochastic learning automata. In Section III, we present STAR with depth $D = 1$ and prove its optimality properties. In Section IV, we present STAR with depth $D > 1$ and prove its optimality properties. In Section V, we present computer simulations to compare the performance of STAR to that of L_{R-P} and $L_{R-\epsilon P}$. Finally, in Section VI, we summarize, present our conclusions, and propose some directions for future research.

II. Fundamentals

In this section, we present the standard mathematical definition of the learning automaton model. This involves the definition of the automaton itself, the environment with which it interacts, the objective of this interaction and the learning method. A discussion of stationary and nonstationary environments is also included.

Environment is defined by a triple $\{\alpha, \beta, c\}$, where

1) $\alpha = \{1, 2, \ldots, r\}$ is the set of actions (input to the environment);
2) $\beta = \{0, 1\}$ is the set of responses (output of the environment);
3) $c = \{c_1, c_2, \ldots, c_r\}$ is an unknown penalty probability set.

Automaton is defined by a quintuple $\{\Phi, \alpha, \beta, \mathbf{F}(., ., .), \mathbf{G}(.)\}$, where

1) $\Phi = \{1, 2, \ldots, s\}$ is the set of the internal states;
2) $\alpha = \{1, 2, \ldots, r\}$ is the set of actions (output of the automaton);
3) $\beta = \{0, 1\}$ is the set of responses (input to the automaton);
4) $\mathbf{F}(., ., .)$: $\Phi * \alpha * \beta \mapsto \Phi$ is the state transition mechanism according to which the next state is chosen (depending on the current state and the environment response);
5) $\mathbf{G}(.)$: $\Phi \mapsto \alpha$ is the action selection mechanism according to which the next action is chosen (depending on the current state).

At each instant n, the automaton selects randomly [according to the action probability vector $p(n)$] an action $\alpha(n) = i$ from the finite action set α. The probability that the automaton selects action i, at time n is the action probability $p_i(n) = \Pr[a(n) = i]$; we have $\sum_{i=1}^{r} p_i(n) = 1 \ \forall n$. The environment responds with $\beta(n)$; when the response is favorable (reward) $\beta(n) = 0$, when it is unfavorable (penalty) $\beta(n) = 1$. The environment response to action i is chosen according to the unknown penalty probability $c_i = \Pr[\beta(n) = 1 | a(n) = i] \ \forall i$. Thus, the environment is characterized by the set of penalty probabilities $c = \{c_1, \ldots, c_r\}$. The environment reward probability is $d_i = 1 - c_i, i = 1, \ldots, r$. The environment penalty probabilities $\{c_i\}$ are unknown to the automaton.

It is desirable that the automaton selects the action associated with the minimum penalty probability $c^* = \min_i\{c_i\}$. Automaton performance is usually evaluated by the average cost for a given action probability vector: $M(n) = E[\beta(n)|p(n)] = \Pr[\beta(n) = 1|p(n)] = \sum_{i=1}^{r} \Pr[\beta(n) = 1|\alpha(n) = i] p_i(n) = \sum_{i=1}^{r} c_i p_i(n)$. Thus, the action a^* producing the c^* is the best action. With no *a priori* information, the automaton selects actions with equal probability $p_i(n) = 1/r, i = 1, \ldots, r$. This is called a *pure-chance automaton.* Then the average cost is the mean of the penalty probabilities $M_0 = (1/r) \sum_{i=1}^{r} c_i$.

Learning can take place by repeated application of the following procedure: the automaton chooses an action according to the current action probability vector and updates this action probability vector according to the environment response. Hopefully, this procedure leads to selection of the best action or, at least, reduction of the cost $M(n)$. Formally, the action probability vector at time n, $p(n)$, is updated by a learning algorithm $\mathbf{T}$: $p(n+1) = \mathbf{T}(p(n), \alpha(n), \beta(n))$. The design problem is to specify $\mathbf{T}$ in such a manner that as the updating process evolves, the automaton learns more about the environment, and improves its performance [i.e., reduces $M(n)$]. For example, a learning automaton that asymptotically behaves better than a pure chance automaton will in the limit have average cost $\lim_{n\to\infty} E[M(n)] < M_0$. Such an automaton is called *expedient*. Similarly, a learning automaton is said to be *optimal* if $\lim_{n\to\infty} E[M(n)] = c^*$. Optimality implies that asymptotically the action with the lowest penalty probability is selected with probability one.

Optimality is desirable in *stationary* environments, but a suboptimal performance may be preferable in *nonstationary* ones. An environment is called nonstationary if the penalty probabilities vary with time. This situation occurs frequently in applications. For instance, in a control problem the characteristics of the plant may change in time, so that different costs are associated

[1]This form of ϵ-optimality of STAR$^{(D)}$ is proven in exactly the same way for the two-action and multiaction case, using relatively simple mathematical tools, such as the theory of finite Markov chains. On the contrary, the analysis of VSSA requires more delicate arguments and use of the theory of stochastic difference equations. Especially for the case of $L_{R-\epsilon P}$, an approximation argument is required [4, pp. 166–168].

at different times with the same action. Similar situations occur in telephone and computer network routing. The importance of adaptation becomes even more obvious in such situations. The automaton must not only learn the characteristics of the environment, but also "forget" old characteristics and acquire new ones, in response to the time-varying situation. It is by now well understood [4, pp. 227–279] that an optimal automaton may be too rigid to accommodate such requirements. In particular, an optimal automaton may either get locked in an action which is originally optimal but later becomes pessimal. On the other hand, the automaton may be able to respond to changes in the environment but not sufficiently quickly, because it is too heavily committed to a previously optimal action. It has been found [4, pp. 227–279] that in such cases, ϵ-*optimal* automata are better able to respond to a changing environment. An ϵ-optimal automaton is one which satisfies $\lim_{n\to\infty} E[M(n)] < c^* + \epsilon$, with $\epsilon > 0$.

III. STAR$^{(1)}$

In this section, we present the STAR automaton with depth $D = 1$, which we denote by STAR$^{(1)}$. The general case of STAR with arbitrary depth will be presented in the following section.

As mentioned earlier, the action set is $\alpha = \{1, \ldots, r\}$ and the environment response set is $\beta = \{0, 1\}$ (reward and penalty). The automaton can be in any of $r + 1$ states, $\{0, 1, \ldots, r\}$. The state transition and action selection mechanisms are illustrated in Fig. 1(a). The star-shaped structure which gives STAR its name, is clearly illustrated in Fig. 1.

When the automaton is in state i, it performs action i with probability 1 (for $i = 1, 2, \ldots, r$). Therefore, each state is "committed" to a corresponding action, except for state 0 which is a special, "neutral" state: when in state 0, the automaton chooses any of the r actions equiprobably. The action selection mechanism described above can be summarized by the action selection probability G: $\Phi \to \alpha$ as follows:

$$G_{ij} \doteq \Pr[\alpha(n) = j|\Phi(n) = i] = 1 \qquad i = 1, \ldots, r, \quad j = i \tag{1}$$

$$G_{ij} \doteq \Pr[\alpha(n) = j|\Phi(n) = i] = \frac{1}{r} \qquad i = 0, \quad j = 1, \ldots, r. \tag{2}$$

(All probabilities not listed above are equal to zero.) To evaluate the expediency and optimality of the automaton, we need to know the action probabilities $p_j(n)$ $(j = 1, \ldots, r)$ written in vector form as $p(n) \doteq [p_1(n) \cdots p_r(n)]$. We also define the probability of being at state i at time n: $\pi_i(n) \doteq \mathrm{Prob}\,[\Phi(n) = i]$ $(i = 0, 1, \ldots, r)$; written in vector form as $\pi(n) \doteq [\pi_0(n)\ \pi_1(n) \cdots \pi_r(n)]$. We have the following relationship between action probabilities and state probabilities:

$$p(n) = \pi(n) \cdot G. \tag{3}$$

Hence, both the learning behavior and optimality properties depend on the state probabilities $\pi(n)$, which in turn depend on the state transition mechanism defined by the probabilities F: $\Phi * \alpha * \beta \to \Phi$, as follows:

$$F_{ijk,\,l} \doteq \Pr[\Phi(n+1) = l|\Phi(n) = i,\, \alpha(n) = j,\, \beta(n) = k] \qquad i, l = 0, 1, \ldots, r \quad j = 1, \ldots, r \quad k = 0, 1. \tag{4}$$

These probabilities depend on the current state, action, and response. We will present several possible choices of F for all of which the state process $\Phi(n)$ is an ergodic Markov chain with state transition matrix P, where $P_{il} \doteq \Pr[\Phi(n+1) = l|\Phi(n) = i]$ $(i, l = 0, 1, \ldots, r)$. Hence, $\lim_{n\to\infty} \pi(n) = \pi$, where π is a *unique* equilibrium probability vector. We now proceed to define F, distinguishing four cases.

A. Deterministic Reward–Deterministic Penalty

In this case, both reward and penalty cause deterministic state transitions, according to the following.

1) When in state 0 and chosen action is i $(i = 1, \ldots, r)$, if rewarded go to state i with probability 1 (with probability one)

$$F_{0i0,\,i} = 1, \quad F_{0i0,\,j} = 0 \qquad i = 1, \ldots, r, \quad j = 0, 1, \ldots, r \quad j \neq i \tag{5}$$

if punished, stay in state 0 with probability 1

$$F_{0i1,\,0} = 1, \quad F_{0i1,\,j} = 0 \qquad i, j = 1, \ldots, r. \tag{6}$$

2) When in state i, $i \neq 0$ and chosen action is i $(i = 1, \ldots, r)$, if rewarded stay in state i with probability 1

$$F_{ii0,\,i} = 1, \quad F_{ii0,\,j} = 0 \qquad i = 1, \ldots, r \quad j = 0, 1, \ldots, r \quad j \neq i \tag{7}$$

if punished, go to state 0 with probability 1

$$F_{ii1,\,0} = 1, \quad F_{ii1,\,j} = 0, \qquad i, j = 1, \ldots, r. \tag{8}$$

From F we can compute the equilibrium state probabilities π and action probabilities p and prove the expediency of the automaton. Here, we only present the results of our analysis; detailed derivations are given in the Appendix. The nonzero elements of P turn out to be (for $i = 1, \ldots, r$)

$$P_{00} = \frac{1}{r}\sum_{i=1}^{r} c_i, \quad P_{0i} = \frac{1-c_i}{r} \qquad P_{i0} = c_i, \quad P_{ii} = 1 - c_i \tag{9}$$

all the other elements of P are zero. From (9) it is obvious that $P_{ii} > 0$ for $i = 0, 1, \ldots, r$. Furthermore, it is easy to check that $P^2 > 0$. Hence, the state process $\Phi(n)$ is irreducible, aperiodic and, as a consequence, ergodic [4]. From P we can compute the state probabilities π, which turn out to be

$$\pi_0 = \frac{r}{\sum_{i=1}^{r} \frac{1}{c_i}}, \quad \pi_i = \frac{r}{\sum_{j=1}^{r} \frac{1}{c_j}} \cdot \frac{1-c_i}{c_i} \qquad i = 1, 2, \ldots, r. \tag{10}$$

Now, taking the limit of (3) as $n \to \infty$, we obtain the limit action probabilities as $p = \pi G$ and finally find

$$p_j = \frac{\frac{1}{c_j}}{\sum_{i=1}^{r} \frac{1}{c_i}} \qquad j = 1, \ldots, r. \tag{11}$$

It is easy to compute the limiting average cost. We have

$$M_1 = \sum_{j=1}^{r} c_j \cdot p_j = \sum_{j=1}^{r} c_j \cdot \frac{\frac{1}{c_j}}{\sum_{i=1}^{r} \frac{1}{c_i}} = \sum_{j=1}^{r} \frac{1}{\sum_{i=1}^{r} \frac{1}{c_i}} = \frac{r}{\sum_{i=1}^{r} \frac{1}{c_i}}. \tag{12}$$

Since in the limit the action probabilities of STAR$^{(1)}$ are the same as those of the variable structure L_{R-P} automaton, which is known to be expedient, STAR$^{(1)}$ with deterministic reward and deterministic penalty is also expedient.

B. Deterministic Reward–Probabilistic Penalty

In this case, reward causes deterministic state transitions, but penalty causes probabilistic state transitions, according to the following rules, which make use of the number δ, with $0 < \delta < 1$.

1) When in state 0 and chosen action is i ($i = 1, \ldots, r$), if rewarded go to state i with probability 1

$$F_{0i0,\,i} = 1, \quad F_{0i0,\,j} = 0 \qquad i = 1, \ldots, r \quad j = 0, 1, \ldots, r \quad j \neq i \tag{13}$$

but if punished, go to state i with probability δ or stay in state 0 with probability $1 - \delta$

$$F_{0i1,\,i} = \delta, \quad F_{0i1,\,0} = 1 - \delta, \quad F_{0i1,\,j} = 0 \qquad i, j = 1, \ldots, r \quad j \neq i. \tag{14}$$

2) When in state i, $i \neq 0$ and chosen action is i ($i = 1, \ldots, r$), if rewarded stay in state i with probability 1

$$F_{ii0,\,i} = 1, \quad F_{ii0,\,j} = 0 \qquad i = 1, \ldots, r \quad j = 0, 1, \ldots, r \quad j \neq i \tag{15}$$

if punished, stay in state i with probability δ, or go to state 0 with probability $1 - \delta$

$$F_{ii1,\,i} = \delta, \quad F_{ii1,\,0} = 1 - \delta, \quad F_{ii1,\,j} = 0 \qquad i, j = 1, \ldots, r, \quad j \neq i. \tag{16}$$

As in the previous subsection, from F we compute P. The nonzero elements of P are

$$P_{00} = \frac{1-\delta}{r} \cdot \sum_{i=1}^{r} c_i, \quad P_{0i} = \delta \cdot \frac{c_i}{r} + \frac{1 - c_i}{r} \tag{17}$$

$$P_{i0} = (1 - \delta) \cdot c_i, \quad P_{ii} = 1 - c_i + \delta \cdot c_i \tag{18}$$

for $i = 1, \ldots, r$; all the other elements of P are zero. By the same arguments mentioned previously, $\Phi(n)$ is ergodic. Hence, π and p are determined by P; in particular π turns out to be

$$\pi_0 = \frac{r}{\sum_{i=1}^{r} \frac{1}{\hat{c}_i}}, \quad \pi_i = \frac{r}{\sum_{j=1}^{r} \frac{1}{\hat{c}_j}} \cdot \frac{1 - \hat{c}_i}{\hat{c}_i} \qquad i = 1, 2, \ldots r \tag{19}$$

where $\hat{c}_i = (1 - \delta) \cdot c_i$ (for $i = 1, \ldots, r$). Hence, (19) has the same form as (11), but in place of c_i we now have $\hat{c}_i$. As in the previous case, we find

$$p_j = \frac{\frac{1}{\hat{c}_j}}{\sum_{i=1}^{r} \frac{1}{\hat{c}_i}} \qquad j = 1, \ldots r \tag{20}$$

$$M_1^{\delta} = \frac{r}{\sum_{i=1}^{r} \frac{1}{\hat{c}_i}} = (1 - \delta) \cdot \frac{r}{\sum_{i=1}^{r} \frac{1}{c_i}} < M_1. \tag{21}$$

Hence, for any $\delta > 0$, STAR$^{(1)}$ with deterministic reward and probabilistic penalty has superior performance to L_{R-P}, as well as to STAR$^{(1)}$ with deterministic reward and deterministic penalty. From this, it follows immediately that it is also expedient.[2]

C. Probabilistic Reward–Deterministic Penalty

In this case, reward causes probabilistic state transitions, but penalty causes deterministic state transitions, according to the following rules, which make use of the number ϵ, with $0 < \epsilon < 1$.

1) When in state 0 and chosen action is i ($i = 1, \ldots, r$), if rewarded go to state i with probability $1 - \epsilon$ or stay in state 0 with probability ϵ

$$F_{0i0,\,i} = 1 - \epsilon, \quad F_{0i0,\,0} = \epsilon, \quad F_{0i0,\,j} = 0, \qquad i, j = 1, \ldots, r \quad j \neq i \tag{22}$$

but if punished, stay in state 0 with probability 1

$$F_{0i1,\,0} = 1, \quad F_{0i1,\,j} = 0 \qquad i, j = 1, \ldots, r. \tag{23}$$

2) When in state i, $i \neq 0$ and chosen action is i ($i = 1, \ldots, r$), if rewarded stay in state i with probability $1 - \epsilon$ or go to state 0 with probability ϵ

$$F_{ii0,\,i} = 1 - \epsilon, \quad F_{ii0,\,0} = \epsilon, \quad F_{ii0,\,j} = 0 \qquad i, j = 1, \ldots, r, \quad j \neq i \tag{24}$$

if punished, go to state 0 with probability 1

$$F_{ii1,\,0} = 1, \quad F_{ii1,\,j} = 0, \qquad i, j = 1, \ldots, r. \tag{25}$$

[2]From (21) it may appear that for $\delta = 1$ we have $M_1^{\delta} = 0$. This is not the case; when $\delta = 1$, it is no longer true that $P^2 > 0$. In fact, we have $P_{j0} = 0$, $\forall\, j$, and $P_{ii} = 1$ for $\forall\, i \neq 0$; hence, states $i = 1, \ldots, r$ are absorbing states and the state process $\Phi(n)$ is not ergodic. Intuitively, this follows from the fact that when $\delta = 1$, no penalty is applied [consider (16)]. The upshot of all this is that the automaton has no steady-state probabilities π and average cost M_1^{δ} is not well defined. As a practical matter, the choice of $\delta = 1$ must be avoided.

As in the previous subsection, from F we compute P. The nonzero elements of P are

$$P_{00} = \frac{1}{r}\sum_{i=1}^{r} c_i + \epsilon \cdot \frac{1}{r}\sum_{i=1}^{r}(1-c_i)$$

$$P_{0i} = (1-\epsilon)\cdot\frac{1-c_i}{r} \tag{26}$$

$$P_{i0} = c_i + \epsilon\cdot(1-c_i), \quad P_{ii} = (1-\epsilon)\cdot(1-c_i) \tag{27}$$

for $i = 1, \ldots, r$; all the other elements of P are zero. From P we infer that $\Phi(n)$ is ergodic and compute π and p

$$\pi_0 = \frac{r}{\sum_{i=1}^{r}\frac{1}{\overline{c}_i}}, \quad \pi_i = \frac{r}{\sum_{j=1}^{r}\frac{1}{\overline{c}_j}}\cdot\frac{1-\overline{c}_i}{\overline{c}_i}$$
$$i = 1, 2, \ldots, r \tag{28}$$

with $\overline{c}_i = c_i + \epsilon\cdot(1-c_i) > c_i$ (for $i = 1, \ldots, r$).

Similarly

$$p_j = \frac{\frac{1}{\overline{c}_j}}{\sum_{i=1}^{r}\frac{1}{\overline{c}_i}} \qquad j = 1, \ldots, r \tag{29}$$

and the limiting average cost is

$$M_1^{\epsilon} = \frac{r}{\sum_{i=1}^{r}\frac{1}{\overline{c}_i}}. \tag{30}$$

Since $\overline{c}_i > c_i$, $i = 1, \ldots, r$, we see that $M_1^{\epsilon} > M_1$, and so STAR$^{(1)}$ with probabilistic reward and deterministic penalty performs worse than either the L_{R-P} or STAR$^{(1)}$ with deterministic reward and deterministic penalty.

D. Probabilistic Reward–Probabilistic Penalty

This is the most general case: both reward and penalty cause probabilistic state transitions as follows.

1) When in state 0 and action is i ($i = 1, \ldots, r$), if rewarded go to state i with probability $1-\epsilon$ or stay in state 0 with probability ϵ

$$F_{0i0,i} = 1-\epsilon, \quad F_{0i0,0} = \epsilon, \quad F_{0i0,j} = 0$$
$$i, j = 1, \ldots, r \quad j \neq i \tag{31}$$

but if punished, go to state i with probability δ, or stay in state 0 with probability $1-\delta$

$$F_{0i1,i} = \delta, \quad F_{0i1,0} = 1-\delta, \quad F_{0i1,j} = 0$$
$$i, j = 1, \ldots, r \quad j \neq i. \tag{32}$$

2) When in state i, $i \neq 0$ and action is i ($i = 1, \ldots, r$), if rewarded stay in state i with probability $1-\epsilon$ or go to state 0 with probability ϵ

$$F_{ii0,i} = 1-\epsilon, \quad F_{ii0,0} = \epsilon, \quad F_{ii0,j} = 0$$
$$i, j = 1, \ldots, r \quad j \neq i \tag{33}$$

if punished, go to state 0 with probability $1-\delta$, or stay in state i with probability δ

$$F_{ii1,i} = \delta, \quad F_{ii1,0} = 1-\delta, \quad F_{ii1,j} = 0$$
$$i, j = 1, \ldots, r, \quad j \neq i. \tag{34}$$

As in the previous subsection, from F we compute π and p; The nonzero elements of P are

$$P_{00} = \frac{1-\delta}{r}\cdot\sum_{i=1}^{r} c_i + \frac{\epsilon}{r}\cdot\sum_{i=1}^{r}(1-c_i)$$

$$P_{0i} = \delta\cdot\frac{c_i}{r} + (1-\epsilon)\cdot\frac{1-c_i}{r} \tag{35}$$

$$P_{i0} = (1-\delta)\cdot c_i + \epsilon\cdot(1-c_i)$$

$$P_{ii} = \delta\cdot c_i + (1-\epsilon)\cdot(1-c_i) \tag{36}$$

for $i = 1, \ldots, r$; all the other elements of P are zero. By the same arguments as discussed previously, it is seen that $\Phi(n)$ is ergodic; its equilibrium state probabilities π turn out similar to (11) but cannot be written conveniently in closed-form.

We observe that the previous three forms of F are special cases of this one: deterministic reward–deterministic penalty uses $\delta = 0$, $\epsilon = 0$, deterministic reward–probabilistic penalty uses $0 < \delta < 1$, $\epsilon = 0$, probabilistic reward–deterministic penalty uses $\delta = 0$, $0 < \epsilon < 1$.

IV. STAR$^{(D)}$

In this section, we present the STAR$^{(D)}$ automaton with arbitrary depth D. The action set α and the response set β are the same as in the previous section. However, STAR$^{(D)}$ has more states than STAR$^{(1)}$ and a somewhat different labeling convention is used. States are numbered by pairs of integers, as follows.

1) The state $(0, 0)$ is the neutral state (all actions are equiprobable).

2) The state (i, d) is the dth state committed to action i. Hence, the index i runs from 1 to r and the index d runs from 1 to D.

This numbering of the states corresponds to the star-shaped structure of Fig. 1(b). States are partitioned into r sets of D states each, each set forming a *branch* of the star, each branch being committed to one of the r possible actions. Every time the automaton chooses action i and is rewarded, it goes to a state deeper into the ith branch; when it is punished it moves toward the neutral state $(0, 0)$, where every action is equiprobable. Thus, the operation of each branch of the automaton state diagram resembles that of a stack.

The action selection mechanism is the same as for STAR$^{(1)}$ and is described by G: $\Phi \to \alpha$ [note that the state set Φ is different from that of STAR$^{(1)}$]

$$G_{(i,d),j} \doteq \Pr[\alpha(n) = j | \Phi(n) = (i, d)] = 1$$
$$i = j = 1, \ldots, r, \quad d = 1, \ldots, D \tag{37}$$

$$G_{(0,0),j} \doteq \Pr[\alpha(n) = j | \Phi(n) = (0, 0)] = \frac{1}{r}$$
$$j = 1, \ldots, r. \tag{38}$$

Action and state probabilities are defined in the same manner as for STAR$^{(1)}$; once again Φ is different from that of STAR$^{(1)}$

$$p_j(n) \doteq \Pr[\alpha(n) = j] \qquad j = 1, \ldots, r \tag{39}$$

in vector form, $p(n) \doteq [p_1(n) \cdots p_r(n)]$

$$\pi_{(i,d)}(n) \doteq \Pr[\Phi(n) = (i, d)] \quad (i, d) = (0, 0) \text{ or } i = 1, \ldots, r,$$
$$d = 1, \ldots, D \tag{40}$$

in vector form, $\pi(n) \doteq [\pi_{(0,0)}(n)\ \pi_{(1,1)}(n) \cdots \pi_{(r,D)}(n)]$. The following relationship holds between action probabilities and state probabilities:

$$p_j(n) = \sum_{\forall (i,d)} \pi_{(i,d)}(n) \cdot G_{(i,d),j}. \tag{41}$$

In the following paragraphs, the process $\Phi(n)$ will always be ergodic; hence, it has a unique equilibrium probability vector $\pi \doteq [\pi_{(0,0)}\ \pi_{(1,1)} \cdots \pi_{(r,D)}]$, where $\pi_{(i,d)}$ is defined by

$$\pi_{(i,d)} \doteq \lim_{n\to\infty} \pi_{(i,d)}(n), \quad (i,d) = (0,0) \text{ or } i = 1, \ldots, r \quad d = 1, \ldots, D. \tag{42}$$

The state transition mechanism is defined by the probabilities F: $\Phi * \alpha * \beta \to \Phi$, where

$$F_{(i,d)jk,(i',d')} \doteq \Pr[\Phi(n+1) = (i',d') | \Phi(n) = (i,d), \alpha(n) = j, \beta(n) = k]. \tag{43}$$

The most general F used is characterized by probabilistic reward and probabilistic penalty with $0 < \delta < 1$ and $0 < \epsilon < 1$ (similar to the case of Section III-D). We show below only the nonzero elements of F.

1) When in state $(0,0)$ and chosen action is i, if rewarded go to state $(i,1)$ with probability $1-\epsilon$ or stay in state $(0,0)$ with probability ϵ

$$F_{(0,0)i0,(i,1)} = 1-\epsilon, \quad F_{(0,0)i0,(0,0)} = \epsilon \quad i = 1, \ldots, r \tag{44}$$

but, if punished, go to state $(i,1)$ with probability δ or stay in state $(0,0)$ with probability $1-\delta$

$$F_{(0,0)i1,(i,1)} = \delta, \quad F_{(0,0)i1,(0,0)} = 1-\delta \quad i = 1, \ldots, r. \tag{45}$$

2) When in state $(i,1)$ $i = 1, 2, \ldots, r$, and chosen action is i, if rewarded go to state $(i,2)$ with probability $1-\epsilon$ or go to state $(0,0)$ with probability ϵ

$$F_{(i,1)i0,(i,2)} = 1-\epsilon, \quad F_{(i,1)i0,(0,0)} = \epsilon \quad i = 1, \ldots, r \tag{46}$$

but, if punished, go to state $(i,2)$ with probability δ or go to state $(0,0)$ with probability $1-\delta$

$$F_{(i,1)i1,(i,2)} = \delta, \quad F_{(i,1)i1,(0,0)} = 1-\delta \quad i = 1, \ldots, r. \tag{47}$$

3) When in state (i,d) $i = 1, 2, \ldots, r$, $d = 2, \ldots, D-1$ and chosen action is i, if rewarded go to state $(i, d+1)$ with probability $1-\epsilon$ or go to state $(i, d-1)$ with probability ϵ

$$F_{(i,d)i0,(i,d+1)} = 1-\epsilon, \quad F_{(i,d)i0,(i,d-1)} = \epsilon \quad i = 1, \ldots, r \tag{48}$$

but, if punished, go to state $(i, d+1)$ with probability δ or go to state $(i, d-1)$ with probability $1-\delta$

$$F_{(i,d)i1,(i,d+1)} = \delta, \quad F_{(i,d)i1,(i,d-1)} = 1-\delta \quad i = 1, \ldots, r. \tag{49}$$

4) Finally, when in state (i, D), $i = 1, 2, \ldots, r$ and chosen action is i, if rewarded stay in state (i, D) with probability $1-\epsilon$ or go to state $(i, D-1)$ with probability ϵ

$$F_{(i,D)i0,(i,D)} = 1-\epsilon, \quad F_{(i,D)i0,(i,D-1)} = \epsilon \quad i = 1, \ldots, r \tag{50}$$

but, if punished, stay in state (i, D) with probability δ or go to state $(i, D-1)$ with probability $1-\delta$

$$F_{(i,D)i1,(i,D)} = \delta, \quad F_{(i,D)i1,(i,D-1)} = 1-\delta \quad i = 1, \ldots, r. \tag{51}$$

All elements of F not listed above, are taken to be equal to zero.

Using the above values of F, G, and c probabilities, we obtain a state transition probability matrix $P^{(D)}$, which determines the convergence properties of STAR$^{(D)}$. Omitting the details of the derivation, we give the final result

$$\begin{aligned}
P^{(D)}_{(0,0),(0,0)} &= \frac{1}{r}\sum_{i=1}^{r}((1-c_i)\cdot\epsilon + c_i\cdot(1-\delta)) \\
P^{(D)}_{(0,0),(i,1)} &= \frac{1}{r}((1-c_i)\cdot(1-\epsilon) + c_i\cdot\delta) \\
& \quad i = 1, \ldots, r \\
P^{(D)}_{(i,1),(i,2)} &= (1-c_i)\cdot(1-\epsilon) + c_i\cdot\delta \\
P^{(D)}_{(i,1),(0,0)} &= (1-c_i)\cdot\epsilon + c_i\cdot(1-\delta) \\
& \quad i = 1, \ldots, r \\
P^{(D)}_{(i,d),(i,d+1)} &= (1-c_i)\cdot(1-\epsilon) + c_i\cdot\delta \\
P^{(D)}_{(i,d),(i,d-1)} &= (1-c_i)\cdot\epsilon + c_i\cdot(1-\delta) \\
& \quad i = 1, \ldots, r, \quad d = 2, \ldots, D-1 \\
P^{(D)}_{(i,D),(i,D)} &= (1-c_i)\cdot(1-\epsilon) + c_i\cdot\delta \\
P^{(D)}_{(i,D),(i,D-1)} &= (1-c_i)\cdot\epsilon + c_i\cdot(1-\delta) \\
& \quad i = 1, \ldots, r.
\end{aligned}$$

All the transition probabilities not indicated above are equal to zero. A tedious but straightforward computation shows that $(P^{(D)})^{2D} > 0$. Intuitively, this corresponds to the fact that in $2D$ steps,we can get from any state to any other state, with positive probability. Furthermore, we note that $P^{(D)}_{(0,0),(0,0)} > 0$. Hence, $\Phi(n)$ is irreducible and aperiodic, consequently also ergodic. It follows that there are probability vectors $\pi^{(D)}(n)$, $\pi^{(D)}$, such that $\pi^{(D)}(n+1) = \pi^{(D)}(n)P^{(D)}$, $\pi^{(D)} = \pi^{(D)}P$, $\pi^{(D)} = \lim_{n\to\infty}\pi^{(D)}(n)$. Just like in the STAR$^{(1)}$ case, from $\pi^{(D)}$ we obtain a limiting (equilibrium) action probability vector $p^{(D)} = [p_1^{(D)} \cdots p_r^{(D)}]$.

In the general case, when $0 < \delta < 1$, $0 < \epsilon < 1$, the equilibrium action probabilities cannot be expressed in a compact form. However, we can find compact expressions for special cases.

1) Deterministic Reward–Deterministic Penalty: In this case $\delta = 0$, $\epsilon = 0$. We obtain (see the Appendix)

$$p_i^{(D)} = \frac{\sum_{d=0}^{D}\left(\frac{1-c_i}{c_i}\right)^d}{\sum_{j=1}^{r}\sum_{d=0}^{D}\left(\frac{1-c_j}{c_j}\right)^d} \qquad i = 1, 2, \ldots, r. \tag{52}$$

With a little additional algebra it can be seen that the above action probabilities are very similar to the ones of the Tsetlin automaton $L_{2N,2}$ [4, p. 68].

2) Deterministic Reward–Probabilistic Penalty: In this case $0 < \delta < 1$, $\epsilon = 0$. We obtain (see the Appendix)

$$p_i^{(D)} = \frac{\sum_{d=0}^{D} \left(\frac{1-\hat{c}_i}{\hat{c}_i}\right)^d}{\sum_{j=1}^{r} \sum_{d=0}^{D} \left(\frac{1-\hat{c}_j}{\hat{c}_j}\right)^d} \qquad i = 1, 2, \ldots, r. \tag{53}$$

This is similar to (52) and also to the Tsetlin automaton $L_{2N,2}$ [4, p. 68]. But there is an important difference: in place of c_i we have $\hat{c}_i = (1-\delta)\cdot c_i < c_i$. It must be emphasized that for any value of c_i, the $\hat{c}_i = (1-\delta)\cdot c_i$ can be made smaller than 1/2 by appropriate choice of δ. As will be seen in Theorem 1, it follows that the deterministic reward–probabilistic penalty STAR$^{(D)}$ can be made ϵ-optimal in any environment by appropriate choice of δ.

3) Probabilistic Reward–Deterministic Penalty: In this case $\delta = 0$, $0 < \epsilon < 1$. We obtain (see the Appendix)

$$p_i^{(D)} = \frac{\sum_{d=0}^{D} \left(\frac{1-\bar{c}_i}{\bar{c}_i}\right)^d}{\sum_{j=1}^{r} \sum_{d=0}^{D} \left(\frac{1-\bar{c}_j}{\bar{c}_j}\right)^d} \qquad i = 1, 2, \ldots, r. \tag{54}$$

In all of the above, when $D = 1$, we recover the STAR$^{(1)}$ case. Furthermore, if there is at least one $c_i < 1/2$, from (52) we see that STAR$^{(D)}$ is ϵ-optimal. In fact, we can show a stronger result: by appropriate choice of δ, the *probabilistic reward–deterministic penalty STAR*$^{(D)}$ can be rendered *ϵ-optimal in every environment*; this is proved below.

Theorem 1: Take any environment $\{c_1, c_2, \ldots, c_r\}$. Define $c^* = \min_{1\le i\le r} c_i$.

i) If $c^* < 1/2$, then the deterministic reward–deterministic penalty STAR$^{(D)}$ is ϵ-optimal.

ii) For all $\delta > 1/2$, the deterministic reward–probabilistic penalty STAR$^{(D)}$ is ϵ-optimal.

Proof: i) Assume, without loss of generality, that $c^* = c_1 < 1/2$. Define $g_i = (1-c_i)/c_i$ for $i = 1, \ldots, r$ and note that g_1 is the maximum of $\{g_1, \ldots, g_r\}$ and, in fact, $g_1 > 1$. Choose any $i = 2, \ldots, r$ and take the ratio

$$\frac{p_i^{(D)}}{p_1^{(D)}} = \frac{\sum_{d=0}^{D} \left(\frac{1-c_i}{c_i}\right)^d}{\sum_{d=0}^{D} \left(\frac{1-c_1}{c_1}\right)^d} = \frac{1 + g_i + \cdots + g_i^D}{1 + g_1 + \cdots + g_1^D}$$
$$= \frac{g_1 - 1}{g_i - 1} \cdot \frac{g_i^{D+1} - 1}{g_1^{D+1} - 1}. \tag{55}$$

The first fraction is independent of D and does not affect convergence. There are two cases for the second fraction.

a) If $g_i < 1$, then the numerator tends to -1 and the denominator to ∞, hence the fraction goes to zero.

b) If $g_i > 1$, then the whole fraction tends to $(g_i/g_1)^D$. However, we have assumed that $g_i < g_1$, hence $(g_i/g_1)^D$ goes to zero again.

Hence, $\lim_{D\to\infty}(p_i^{(D)}/p_1^{(D)}) = 0$ (for $i = 2, \ldots, r$). Since for every D we have $\sum_{i=1}^{r} p_i^{(D)} = 1$, it follows that $\lim_{D\to\infty} p_1^{(D)} = 1$, $\lim_{D\to\infty} p_i^{(D)} = 0$ for $i \neq 1$, and the proof of ϵ-optimality is complete.

ii) For every $\delta > 0$ and for $i = 1, 2, \ldots, r$ define $\hat{c}_i(\delta) = (1-\delta)\cdot c_i$ and $\hat{c}^*(\delta) = \min_{1\le i\le r} \hat{c}_i(\delta) = (1-\delta)\cdot c^*$. Furthermore, define $\delta_0 = 1 - (1/2c^*)$. It is easy to check that for all $\delta > \delta_0$ we have $\hat{c}^*(\delta) < 1/2$, which implies [by the same argument used for i)] the ϵ-optimality of the deterministic reward–probabilistic penalty STAR$^{(D)}$. Furthermore, note that in *every* environment $1/2 \ge 1 - (1/2c^*) = \delta_0$. Hence, in *every* environment $\delta > 1/2$ guarantees ϵ-optimality. ■

Hence, STAR$^{(1)}$ is expedient, just like L_{R-P}, and *the deterministic reward/probabilistic penalty* STAR$^{(D)}$ *can be made* ϵ-optimal, just like $L_{R-\epsilon P}$. Furthermore, as will be seen in the next section, the STAR automata are generally faster than the corresponding variable structure automata. They are also easier to implement, requiring no floating point multiplications. Finally, STAR automata are mathematically more tractable, since they can be analyzed by the theory of finite Markov chains; the analysis of $L_{R-\epsilon P}$ behavior requires the use of stochastic difference equations and an approximation argument [4, pp. 166–168].

V. EXPERIMENTS

In this section, we present some computer simulation results to compare the performance of STAR$^{(D)}$ to that of L_{R-P}, $L_{R-\epsilon P}$, and L_{R-I}. The automata are compared for various values of the environment penalty probabilities in a switching environment where the best action changes periodically. An experiment is determined by automaton and environment parameters.

The automata parameters are: depth D for STAR$^{(D)}$; learning rate a for L_{R-P} and L_{R-I}; and learning rates a and b for $L_{R-\epsilon P}$. For every experiment, we have tried ten different values of depth D: $1, 2, \ldots, 10$ and ten different values of learning rate a: 0.01, 0.02, 0.05, 0.10, 0.20, 0.50, 0.70, 0.80, 0.90, 0.99. Regarding the $L_{R-\epsilon P}$ automaton, b must also be specified; we have used: $b = a/10$ and $b = a/5$. We have used ten actions in all experiments.

Regarding the environment parameters, there are ten penalty probabilities c_i, $i = 1, 2, \ldots, 10$. The smallest penalty probability is c_1 for times $n = 1, 2, \ldots, 50, 101, 102, \ldots, 150, \ldots$ and c_2 for times $n = 51, 52, \ldots, 100, 151, 152, \ldots, 200, \ldots$. Hence, in every period of 50 time steps, the learning automaton has to readjust to the best action. We have chosen nine different sets of c_is resulting in nine different experiments. Our c_i choices are summarized in Table I.

The duration of the experiment is 5000 time steps, which means there are 100 switchings of the penalty probabilities. The run length of 5000 steps was sufficient to reach steady-state behavior. Here "steady-state" refers to the time averaged cost incurred by each automaton, which after the first couple thousand steps reaches equilibrium and fluctuates around a mean value. In fact, convergence of average cost occurs within at most 2000

TABLE I
CUMULATIVE RESULTS OF THE NINE EXPERIMENTS. THE COST AVERAGED OVER TEN RUNS, EACH OF 5000 STEPS, IS SHOWN FOR THE BEST STAR$^{(D)}$, L_{R-P}, L_{R-I}, AND $L_{R-\epsilon P}$ AS WELL AS FOR THE STAR$^{(2)}$

Exp.Nr.	1	2	3	4	5	6	7	8	9
c_1	0.04	0.04	0.04	0.04	0.04	0.04	0.44	0.44	0.84
c_2	0.10	0.10	0.10	0.50	0.90	0.90	0.90	0.90	0.90
c_j	0.10	0.50	0.90	0.90	0.50	0.90	0.50	0.90	0.90
STAR$^{(2)}$	0.0905	0.0999	0.0928	0.2739	0.2835	0.2665	0.5101	0.7819	0.8933
STAR$^{(D)}$	**0.0880**	**0.0713**	**0.0705**	**0.2739**	**0.2835**	**0.2665**	**0.5016**	0.7359	0.8917
D	1	7	7	2	2	2	8	6	5
L_{R-P}	0.0896	0.2135	0.2455	0.3534	0.2961	0.3730	0.5145	0.8162	0.8922
a	0.800	0.900	0.990	0.990	0.990	0.990	0.900	0.990	0.990
$L_{R-\epsilon P}$	0.0887	0.0826	0.0785	0.2989	0.4169	0.4029	0.5048	0.7120	0.8906
a	0.100	0.200	0.500	0.200	0.800	0.900	0.050	0.990	0.100
$b = a/10$	0.010	0.020	0.050	0.020	0.080	0.090	0.005	0.099	0.010
$L_{R-\epsilon P}$	0.0895	0.0950	0.0901	0.3216	0.3651	0.3299	0.5313	0.7161	0.8912
a	0.010	0.100	0.500	0.900	0.900	0.990	0.020	0.800	0.100
$b = a/5$	0.002	0.020	0.100	0.180	0.180	0.198	0.004	0.160	0.020
L_{R-I}	0.0882	0.0721	0.0730	0.2800	0.4840	0.4712	0.5023	**0.7082**	**0.8881**
a	0.050	0.100	0.500	0.200	0.200	0.200	0.050	0.100	0.100
M_{th}	0.0870	0.1961	0.2278	0.2786	0.2375	0.2857	0.5159	0.8148	0.8936
M_{opt}	0.0400	0.0400	0.0400	0.0400	0.0400	0.0400	0.4400	0.4400	0.8400

steps; an example of this can be seen in Fig. 2. The action probabilities, on the other hand, periodically change values to follow the environment evolution. We have also run longer simulations (e.g., 10 000 steps, not reported here), but no appreciable change of the average cost was observed. Each 5000-steps run is repeated ten times and the average costs incurred by the STAR$^{(D)}$, L_{R-P}, L_{R-I}, and $L_{R-\epsilon P}$ automata are computed and compared to each other, to the optimal cost M_{opt} and to the theoretical cost M_{th}, where $M_{\text{opt}} = c^*$ and M_{th} are given by (12). In Figs. 3–5, we compare the average cost for all D, a, and b values, for three representative experiments.

The cumulative results can be seen in Table I. Namely, we compare the cost of the best STAR$^{(D)}$ to that of the best L_{R-P}, $L_{R-\epsilon P}$, and L_{R-I}, as well as to theoretical cost M_{th} and optimal cost M_{opt}. Finally, we also list the cost of STAR$^{(2)}$. The minimum cost attained is denoted in bold. We note that in Experiments 1–9, the best STAR$^{(D)}$ outperforms the best L_{R-P}; in Experiments 1–7, it also outperforms the best $L_{R-\epsilon P}$ and L_{R-I}, while in Experiments 8 and 9, STAR$^{(D)}$, $L_{R-\epsilon P}$, and L_{R-I} performance is very close, as can be seen in Fig. 6 (referring to Experiment 8) and in Fig. 7 (referring to Experiment 9). In particular, for Experiment 9, where all the actions are associated with very high penalty probabilities (hence, the choice of action is practically immaterial) *all* automata incur practically the same average cost, between 0.888 and 0.897 (note the reduced scaling of the y-axis in Fig. 7).

In an unknown environment, the optimal value of D, a, and b will not be known in advance. However, in Table I, we see that the value $D = 2$ yields uniformly good results for all environments tested: the cost of STAR$^{(2)}$ is either very close or, in many cases, quite lower than that of M_{th}, and very close to M_{opt}. In short, STAR$^{(2)}$ gives uniformly good performance in a variety of unknown environments.

Regarding the issue of convergence, consider Fig. 8(a)–(c) (referring to Experiment 4) and Fig. 9(a)–(c) (referring to Experiment 6). In each of these cases, we present the evolution of action probability p_1 for 250 steps, encompassing five penalty probability switchings.

In Fig. 8(a), we compare action probabilities p_1 of STAR$^{(2)}$ and L_{R-P} with $a = 0.99$. The optimal p_1 behavior would be the following: in the time intervals 1–50, 101–150, and 201–250, p_1 should be equal to one, since action 1 has the lowest penalty

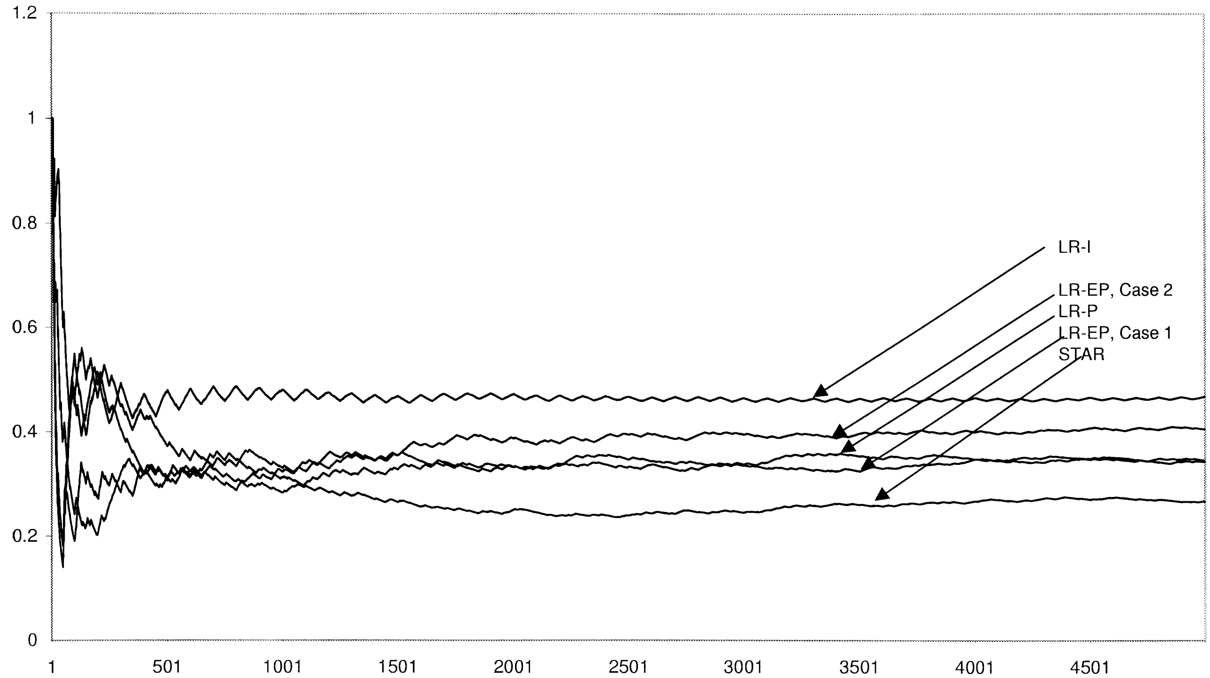


Fig. 2. Average cost (for times 1 to t) plotted versus time t. This refers to one of the ten runs of Experiment 6. Average costs incurred by STAR$^{(2)}$, L_{R-P}, L_{R-I}, and the two examples of $L_{R-\epsilon P}$ are shown. It can be seen that by $t = 2000$, average cost has converged to a steady-state value for all automata.

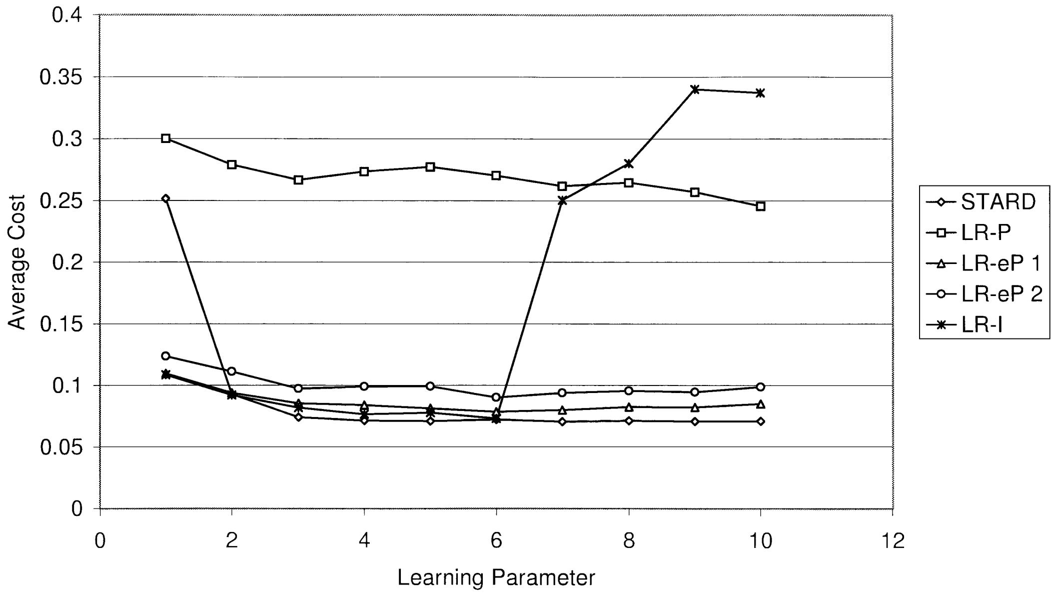


Fig. 3. Experiment 3. Average STAR$^{(D)}$ cost as a function of D; average L_{R-P} and L_{R-I} cost as a function of a; average $L_{R-\epsilon P}$ cost as a function of a, b.

probability; in the time intervals 51–100 and 151–200, p_1 should be zero, since now action 2 has the lowest penalty probability. For STAR, we see that after a few steps p_1 becomes one and stays there until penalty probability switching; shortly afterwards it becomes zero. The same pattern occurs at every penalty probability switching, STAR successfully following the environment. L_{R-P} with $a = 0.99$ has unstable response and is less successful in following the environment. It is worth noting that the high learning rate a (which, we repeat, gave the best results for L_{R-P}) essentially results in behavior resembling that of an FSS automaton. In Fig. 8(b), we compare action probabilities p_1 of STAR$^{(2)}$ and $L_{R-\epsilon P}$ with $a = 0.20$ and $b = 0.02$. Since the STAR p_1 is the same as in Fig. 7, the same conclusions hold. For $L_{R-\epsilon P}$, in this case we have $a = 0.20$ and $b = 0.02$. The small learning rates result in slower response, with the automaton lagging behind environment switchings. In Fig. 8(c), we compare action probabilities p_1 of STAR$^{(2)}$ and $L_{R-\epsilon P}$ with $a = 0.90$ and $b = 0.18$. Again, $L_{R-\epsilon P}$ is slower and less successful than STAR in responding to environment switchings. The high a, b values again result in the $L_{R-\epsilon P}$ displaying near-FSSA behavior. Similar conclusions can be drawn from Fig. 9(a)–(c), which pertain to Experiment 6. Moreover, STAR responds to environment switchings almost instantaneously. Figs. 8 and 9 support the conclusion that STAR responds faster than L_{R-P},

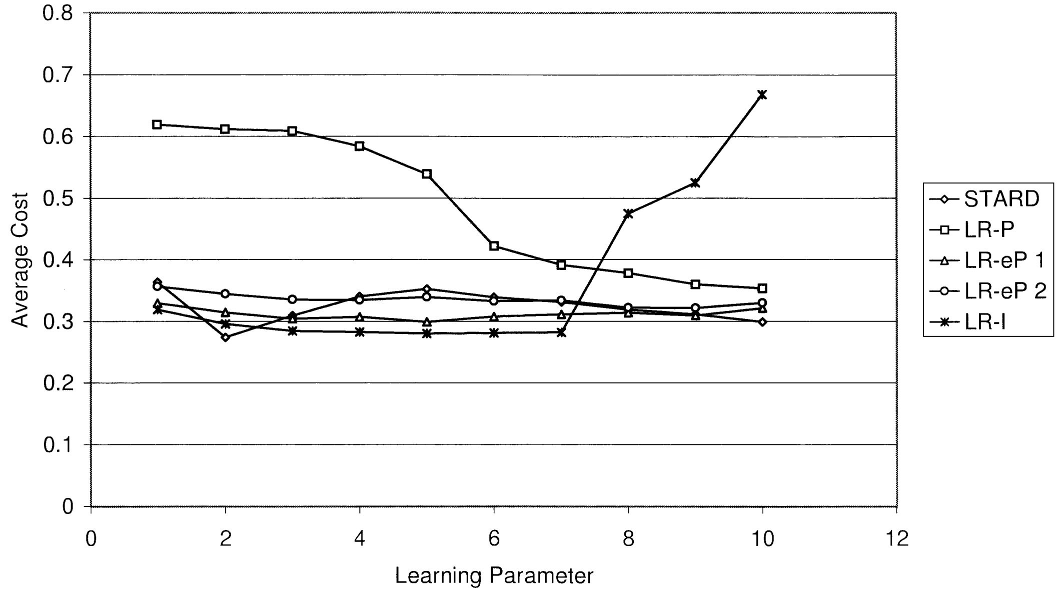


Fig. 4. Experiment 4. Average STAR$^{(D)}$ cost as a function of D; average L_{R-P} and L_{R-I} cost as a function of a; average $L_{R-\epsilon P}$ cost as a function of a, b.

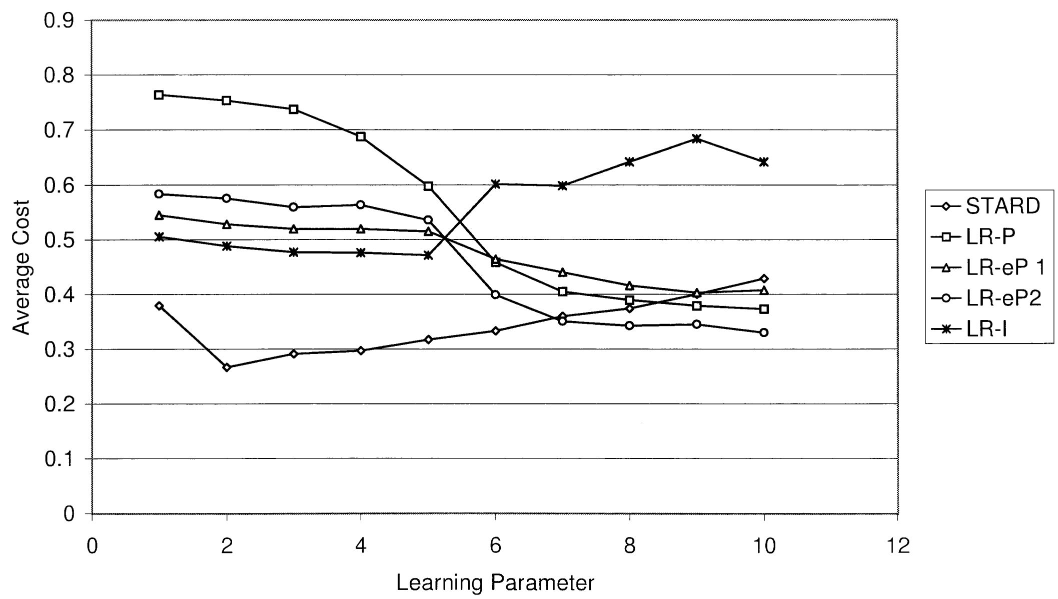


Fig. 5. Experiment 6. Average STAR$^{(D)}$ cost as a function of D; average L_{R-P} and L_{R-I} cost as a function of a; average $L_{R-\epsilon P}$ cost as a function of a, b.

L_{R-I}, and $L_{R-\epsilon P}$ to environment switchings. As we have already seen, in general, it also incurs smaller average cost.

VI. CONCLUSIONS

In this paper, we compared the performance of traditional VSS automata to that of a new class of FSS automata, the so-called STAR$^{(D)}$. Theoretical analysis leads to the following conclusions for a stationary environment. First, STAR$^{(1)}$ with deterministic reward and penalty has the same equilibrium action probabilities and expected cost as L_{R-P}. Second, the introduction of probabilistic penalty makes STAR$^{(1)}$ perform better than L_{R-P}. Finally, when depth D is increased, STAR$^{(D)}$ with deterministic reward and probabilistic penalty can be rendered ϵ-optimal in every environment. We have also performed a number of computer experiments, comparing the performance of L_{R-P}, L_{R-I}, $L_{R-\epsilon P}$, and STAR$^{(D)}$ in nonstationary environments. The conclusion is that in every case, STAR outperforms L_{R-P}; in most cases, it also outperforms $L_{R-\epsilon P}$ and L_{R-I}, except when all actions have very high penalty probabilities, in which case all automata have approximately the same performance. In addition, there is a depth value $D = 2$, which uniformly yields near-optimal results. When the environment parameters are unknown, it is useful to know that for $D = 2$ a uniformly good performance can be achieved. Summarizing, the STAR automata have the following advantages over traditional VSS automata:

1) they generally give better performance;
2) they converge faster;
3) they are simpler to implement.

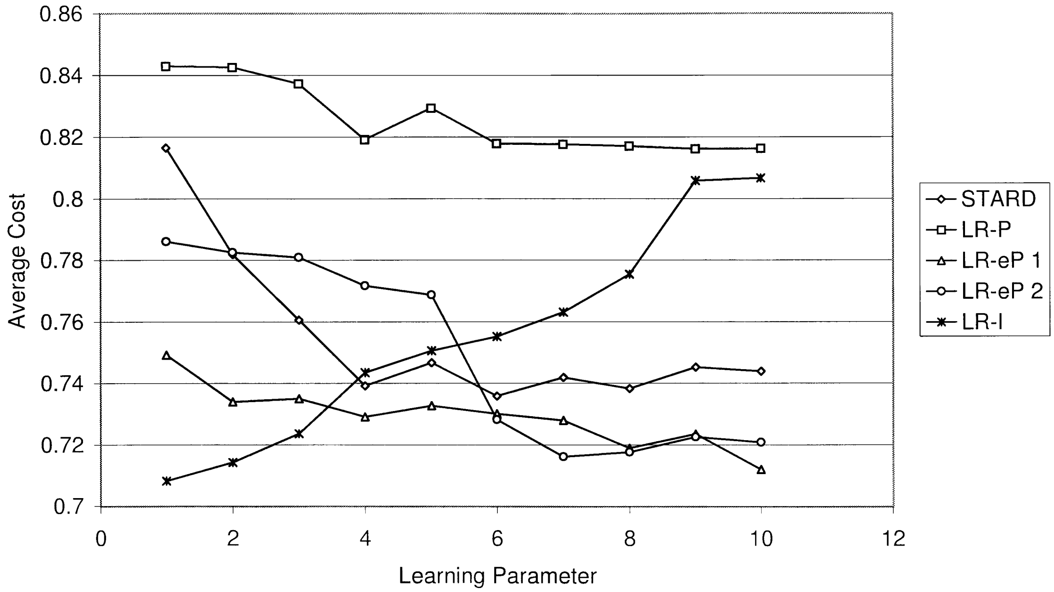


Fig. 6. Experiment 8. Average STAR$^{(D)}$ cost as a function of D; average L_{R-P} and L_{R-I} cost as a function of a; average $L_{R-\epsilon P}$ cost as a function of a, b.

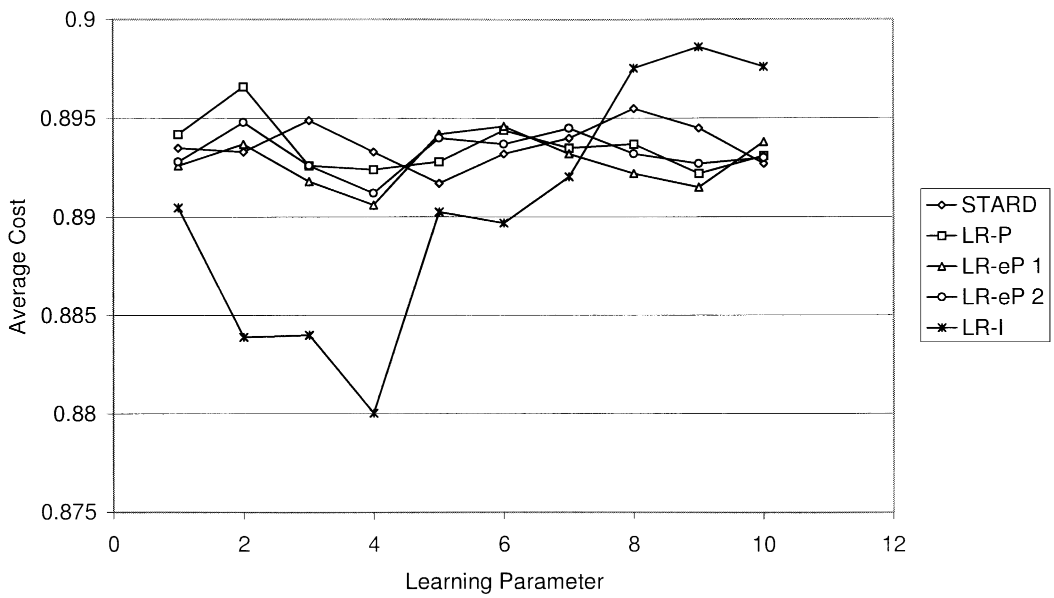


Fig. 7. Experiment 9. Average STAR$^{(D)}$ cost as a function of D; average L_{R-P} and L_{R-I} cost as a function of a; average $L_{R-\epsilon P}$ cost as a function of a, b.

In the future, we intend to compare STAR to discretized VSSA and examine the possibility of implementing estimation and pursuit algorithms by FSSA. We hope our conclusions will stimulate renewed research on FSSA to obtain further high-performance, simple-implementation learning algorithms.

APPENDIX
MATHEMATICAL APPENDIX

A. STAR$^{(1)}$ State Transition Matrix

First, we derive equations for the state transition matrix P for STAR$^{(1)}$. This matrix depends on the parameters δ and ϵ, which lie in the interval $[0, 1]$. We will first derive (35) and (36) for general ϵ and δ; then we will take δ and/or ϵ equal to zero to derive (9), (17), (18), (26), and (27) as special cases.

For instance, let us compute P_{00}, in other words, the probability of transition from state 0 to state 0. We have

$$\begin{aligned} P_{00} &= \Pr(\Phi(n+1)=0|\Phi(n)=0) \\ &= \sum_{i=1}^{r} \Pr(\alpha(n)=i|\Phi(n)=0)\cdot \Pr(\beta(n)=1|\alpha(n)=i) \\ &\quad \cdot \Pr(\Phi(n+1)=0|\beta(n)=1,\, \Phi(n)=0) \\ &\quad + \sum_{i=1}^{r} \Pr(\alpha(n)=i|\Phi(n)=0)\cdot \Pr(\beta(n)=0|\alpha(n)=i) \\ &\quad \cdot \Pr(\Phi(n+1)=0|\beta(n)=0,\, \Phi(n)=0) \\ &= \sum_{i=1}^{r} \frac{1}{r}\cdot c_i \cdot (1-\delta) + \sum_{i=1}^{r} \frac{1}{r}\cdot (1-c_i)\cdot \epsilon. \end{aligned} \tag{56}$$

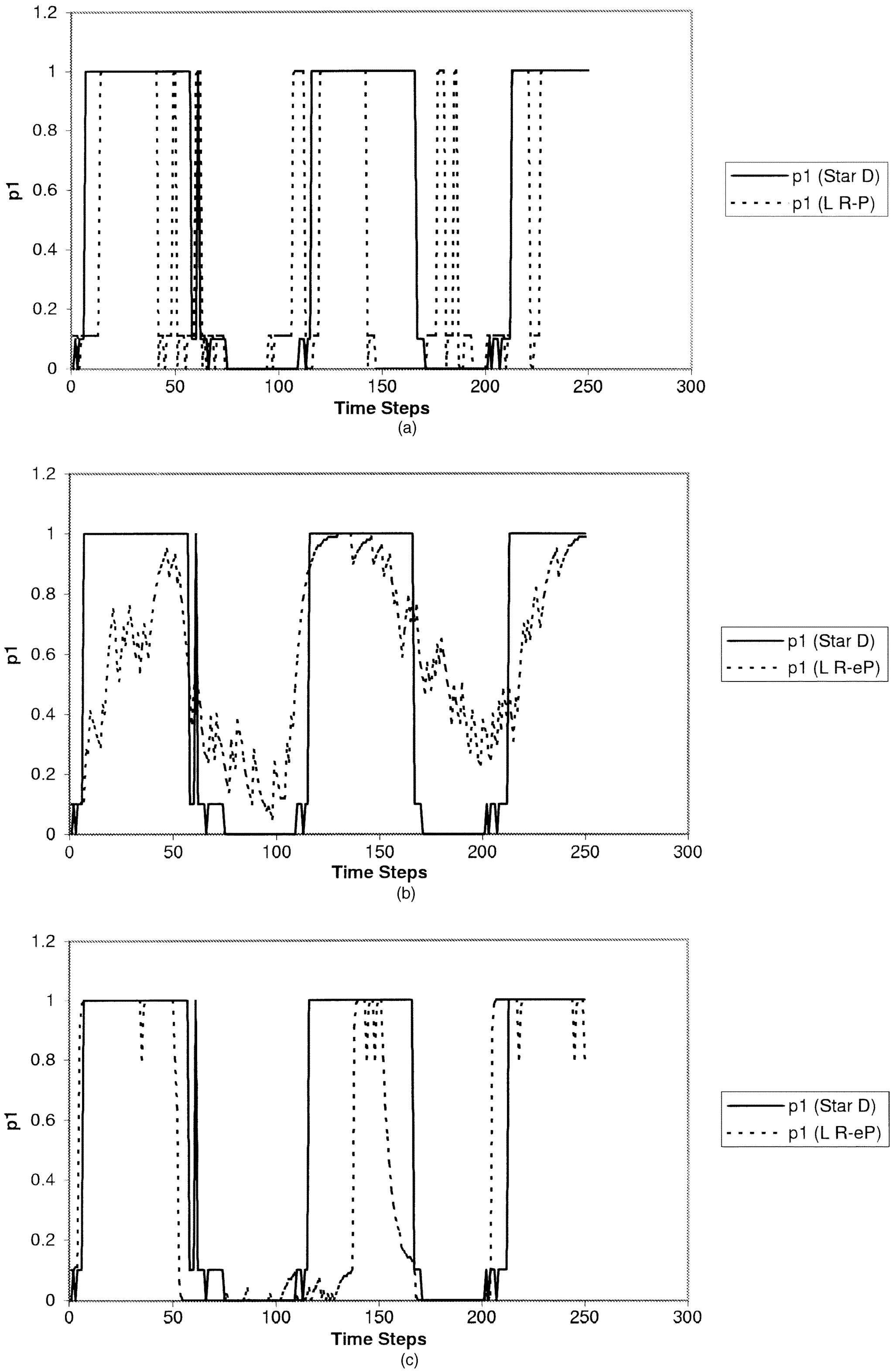


Fig. 8. (a) Experiment 4. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and L_{R-P} $(a = 0.99)$ p_1 probability (dotted line). (b) Experiment 4. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and $L_{R-\epsilon P}$ $(a = 0.20,\ b = 0.02)$ p_1 probability (dotted line). (c) Experiment 4. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and $L_{R-\epsilon P}$ $(a = 0.90,\ b = 0.18)$ p_1 probability (dotted line).

Similarly, for $i = 1, \ldots, r$

$$\begin{aligned} P_{0i} &= \mathrm{Prob}(\Phi(n+1) = i|\Phi(n) = 0) \\ &= \Pr(\alpha(n) = i|\Phi(n) = 0) \cdot \Pr(\beta(n) = 1|\alpha(n) = i) \\ &\quad \cdot \Pr(\Phi(n+1) = i|\beta(n) = 1,\, \Phi(n) = 0) \\ &\quad + \Pr(\alpha(n) = i|\Phi(n) = 0) \cdot \Pr(\beta(n) = 0|\alpha(n) = i) \\ &\quad \cdot \Pr(\Phi(n+1) = i|\beta(n) = 0,\, \Phi(n) = 0) \\ &= \frac{1}{r} \cdot c_i \cdot \delta + \frac{1}{r} \cdot (1 - c_i) \cdot (1 - \epsilon). \end{aligned} \tag{57}$$

Equations (56) and (57) are equivalent to (35). We also have for

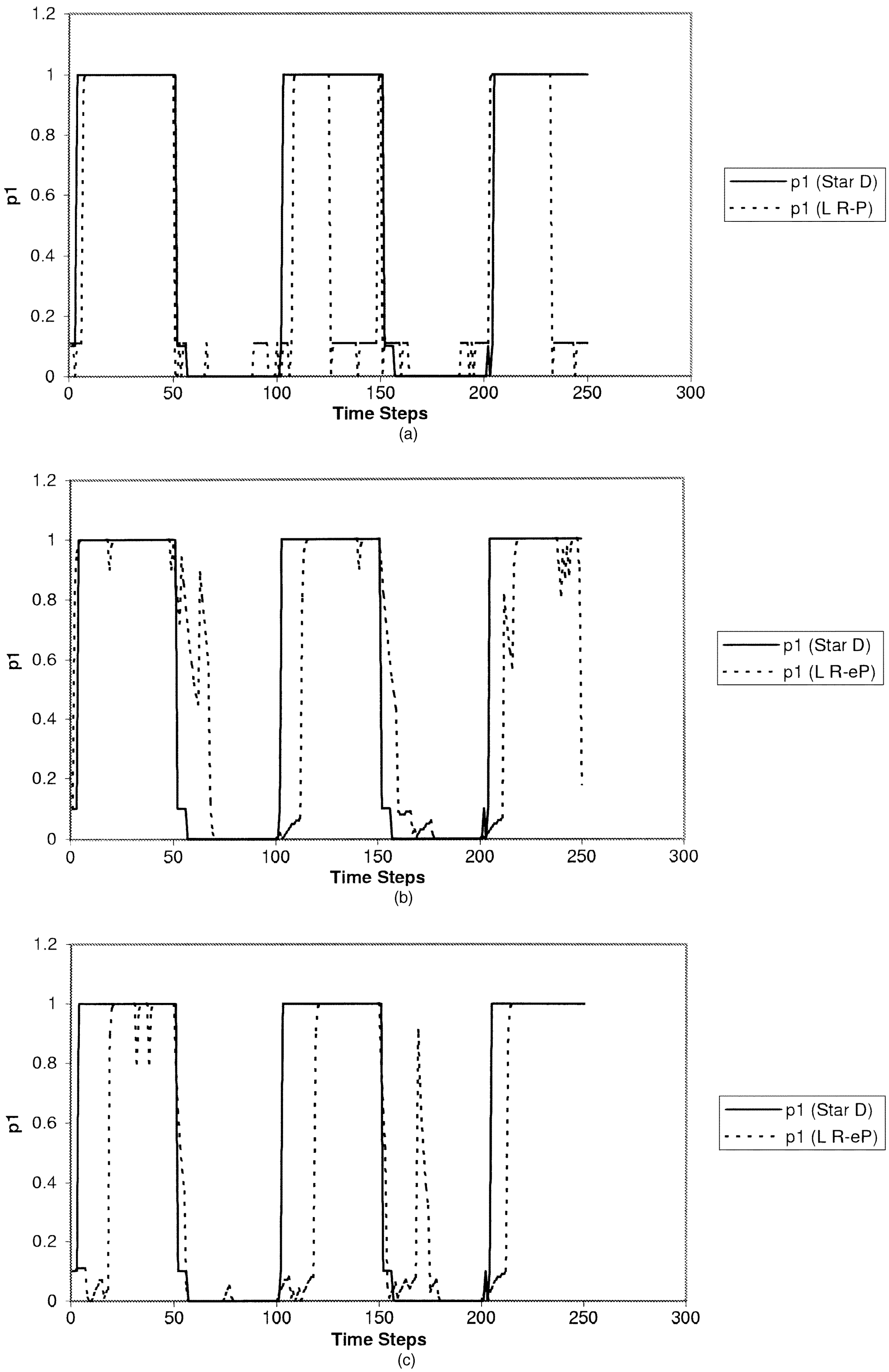


Fig. 9. (a) Experiment 6. 250 time steps profiles of STAR$^{(D)}$ ($D = 2$) p_1 probability (solid line) and L_{R-P} ($a = 0.99$) p_1 probability (dotted line). (b) Experiment 6. 250 time steps profiles of STAR$^{(D)}$ ($D = 2$) p_1 probability (solid line) and $L_{R-\epsilon P}$ ($a = 0.90$, $b = 0.09$) p_1 probability (dotted line). (c) Experiment 6. 250 time steps profiles of STAR$^{(D)}$ ($D = 2$) p_1 probability (solid line) and $L_{R-\epsilon P}$ ($a = 0.990$, $b = 0.198$) p_1 probability (dotted line).

$i = 1, \ldots, r$

$$\begin{aligned} P_{i0} &= \Pr(\Phi(n+1) = 0 | \Phi(n) = i) \\ &= \Pr(\alpha(n) = i | \Phi(n) = i) \cdot \Pr(\beta(n) = 1 | \alpha(n) = i) \\ &\quad \cdot \Pr(\Phi(n+1) = 0 | \beta(n) = 1,\, \Phi(n) = i) \\ &\quad + \Pr(\alpha(n) = i | \Phi(n) = i) \cdot \Pr(\beta(n) = 0 | \alpha(n) = i) \\ &\quad \cdot \Pr(\Phi(n+1) = 0 | \beta(n) = 0,\, \Phi(n) = i) \\ &= 1 \cdot c_i \cdot (1 - \delta) + 1 \cdot (1 - c_i) \cdot \epsilon \end{aligned} \tag{58}$$

and

$$\begin{aligned}P_{ii} &= \Pr(\Phi(n+1)=i|\Phi(n)=i)\\ &= \Pr(\alpha(n)=i|\Phi(n)=i)\cdot\Pr(\beta(n)=1|\alpha(n)=i)\\ &\quad\cdot\Pr(\Phi(n+1)=i|\beta(n)=1,\,\Phi(n)=i)\\ &\quad+\Pr(\alpha(n)=i|\Phi(n)=i)\cdot\Pr(\beta(n)=0|\alpha(n)=i)\\ &\quad\cdot\Pr(\Phi(n+1)=i|\beta(n)=0,\,\Phi(n)=i)\\ &=1\cdot c_i\cdot\delta+1\cdot(1-c_i)\cdot(1-\epsilon).\end{aligned}\tag{59}$$

Equations (58) and (59) are equivalent to (36). Now, letting $\delta = 0$, ϵ arbitrary, from (58) and (59) we obtain (26) and (27); letting $\epsilon = 0$, δ arbitrary, from (58) and (59) we obtain (17) and (18); letting $\delta = 0$, $\epsilon = 0$, from (58) and (59) we obtain (9) and we are done.

B. STAR$^{(1)}$ Equilibrium Probabilities

Next, we obtain closed-form expressions for the equilibrium state and action probabilities of STAR$^{(1)}$ for the case where either δ or ϵ or both are zero.

We first consider the case $\delta = 0$ and $\epsilon = 0$. As discussed in Section III, $\Phi(n)$, the state process, is irreducible and aperiodic, hence ergodic, and it possesses a limiting equilibrium (stationary) probability, called π. To obtain π, we start with the equilibrium equation $\pi = \pi \cdot P$. This matrix equation actually consists of $r+1$ scalar equations. Writing the last r of these explicitly, we get for $i = 1, \ldots, r$

$$\pi_i = \pi_0\cdot\frac{1-c_i}{r}+\pi_i\cdot(1-c_i)\Rightarrow\pi_i=\pi_0\cdot\frac{1}{r}\cdot\frac{1-c_i}{c_i}.$$

This, together with the fact that $\sum_{j=0}^{r}\pi_j = 1$ yields

$$\pi_0\cdot\left(r+\sum_{j=1}^{r}\frac{1-c_j}{c_j}\right)\cdot\frac{1}{r}=1\Rightarrow$$

$$\pi_0=\frac{r}{\sum\limits_{j=1}^{r}\frac{1}{c_i}}\quad \pi_i=\frac{1}{\sum\limits_{j=1}^{r}\frac{1}{c_i}}\cdot\frac{1-c_i}{c_i}\qquad i=1,\ldots,r$$

which are exactly (10). To obtain (11), we observe that action i can only be taken when in state 0 or in state i. Hence, for $i = 1, \ldots, r$, we have

$$p_i=\frac{r}{\sum\limits_{j=1}^{r}\frac{1}{c_j}}\cdot\frac{1}{r}+\frac{1}{\sum\limits_{j=1}^{r}\frac{1}{c_j}}\cdot\frac{1-c_i}{c_i}=\frac{1+\frac{1}{c_i}-1}{\sum\limits_{j=1}^{r}\frac{1}{c_j}}$$

which is exactly (11).

If in place of c_i we use $\hat{c}_i = c_i \cdot (1-\delta)$, then (17) and (18), which define P, take exactly the same form as (9). Hence, the equilibrium probabilities π and the action probabilities p are also of exactly the same form as (10) and (11), except that we have $\hat{c}_i$ in place of c_i. This yields (19) and (20). Similarly, if we use in (26) and (27), $\overline{c}_i = c_i + \epsilon\cdot(1-c_i)$, we obtain (28) and (29) and we are done.

C. STAR$^{(D)}$

We now turn to STAR$^{(D)}$. The computation of the state transition matrix $P^{(D)}$ is the same as for STAR$^{(1)}$ and is not repeated. We now compute the equilibrium probabilities $\pi^{(D)}$. In what follows,. we drop the superscript D for the sake of brevity.

We start with $\delta = 0$, $\epsilon = 0$. $\Phi(n)$, the state process, is irreducible and aperiodic, hence ergodic, and it possesses a limiting equilibrium (stationary) probability π. For the ith branch of the star ($i = 1, \ldots, r$) we obtain terminal conditions

$$\begin{gathered}\pi_{(0,\,0)}\cdot\frac{1-c_i}{r}+\pi_{(i,\,2)}\cdot c_i=\pi_{(i,\,1)}\\ \pi_{(i,\,D-1)}\cdot(1-c_i)+\pi_{(i,\,D)}\cdot(1-c_i)=\pi_{(i,\,D)}\end{gathered}\tag{60}$$

and intermediate conditions

$$\begin{aligned}&\pi_{(i,\,1)}\cdot(1-c_i)+\pi_{(i,\,3)}\cdot c_i=\pi_{(i,\,2)},\ \ldots,\\ &\qquad\pi_{(i,\,D-2)}\cdot(1-c_i)+\pi_{(i,\,D)}\cdot c_i=\pi_{(i,\,D-1)}.\end{aligned}\tag{61}$$

Combining (60) and (61), we get

$$\pi_{(i,\,1)}=\frac{1}{r}\cdot\frac{1-c_i}{c_i}\cdot\pi_{(i,\,0)}\quad\text{and}\quad\pi_{(i,\,d)}=\frac{1-c_i}{c_i}\cdot\pi_{(i,\,d-1)}\qquad d=2,\,\ldots,\,D.\tag{62}$$

From this it follows that $\pi_{(i,d)} = ((1-c_i)/c_i)^d \cdot (\pi_{(i,0)}/r)$ ($d = 1, \ldots, D$). Since $\pi_{(0,0)} + \pi_{(i,1)} + \cdots + \pi_{(r,D)} = 1$, we get $\pi_{(0,0)}\cdot(1+(1/r)\sum_{d=1}^{D}((1-c_1)/c_1)^d + \cdots (1/r)\sum_{d=1}^{D}((1-c_r)/c_r)^d) = 1$ which implies (for $i = 1, \ldots, r, d = 1, \ldots, D$)

$$\pi_{(i,\,d)}=\frac{r}{\sum\limits_{j=1}^{r}\sum\limits_{d=0}^{D}\left(\frac{1-c_j}{c_j}\right)^d}\cdot\left(\frac{1-c_i}{c_i}\right)^d\tag{63}$$

and

$$\pi_{(0,\,0)}=\frac{r}{\sum\limits_{j=1}^{r}\sum\limits_{d=0}^{D}\left(\frac{1-c_j}{c_j}\right)^d}.\tag{64}$$

To obtain the equilibrium probability for action i, add (64) and (63) for $d = 1, \ldots, D$

$$p_i=\pi_{(0,\,0)}\cdot\frac{1}{r}+\sum_{d=1}^{D}\pi_{(i,\,d)}=\frac{\sum\limits_{d=0}^{D}\left(\frac{1-c_j}{c_j}\right)^d}{\sum\limits_{j=1}^{r}\sum\limits_{d=0}^{D}\left(\frac{1-c_j}{c_j}\right)^d}\tag{65}$$

(for $i = 1, \ldots, r$). This yields (52). For the cases $0 < \delta < 1$, $\epsilon = 0$, $\delta = 0$, and $0 < \epsilon < 1$, we use appropriate substitutions [just like for the case STAR$^{(1)}$] and obtain (53) and (54).

REFERENCES

[1] R. C. Atkinson and G. H. Bower, *An Introduction to Mathematical Learning Theory*. New York: Wiley, 1965.
[2] R. R. Bush and F. Mosteller, *Stochastic Models and Learning*. New York: Wiley, 1955.
[3] M. F. Norman, *Markov Processes and Learning Models*. New York: Academic, 1972.
[4] K. Narendra and M. A. L. Thathatchar, *Learning Automata*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[5] M. L. Tsetlin, "On the behavior of finite automata in random media," *Autom. Remote Control*, vol. 22, pp. 1210–1219, 1962.
[6] V. I. Varshavskii and I. P. Vorontsova, "On the behavior of stochastic automata with a variable structure," *Autom. Remote Control*, vol. 24, pp. 327–333, 1964.
[7] A. A. Economides, "Learning automata routing in connection-oriented networks," *Int. J. Commun. Syst.*, vol. 8, pp. 225–237, 1995.

[8] M. N. Howell and T. J. Gordon, "Continuous action reinforcement learning automata and their application to adaptive digital filter design," *Eng. Applicat. Artif. Intell.*, vol. 14, pp. 549–561, 2001.

[9] G. I. Papadimitriou and A. S. Pomportsis, "Learning-automata-based scheduling algorithms for input-queued ATM switches," *Neurocomputing Lett.*, vol. 31, pp. 191–195, 2000.

[10] ——, "On the use of learning automata in medium access control of single-hop lightwave networks," *Comput. Commun.*, vol. 23, pp. 783–792, 2000.

[11] ——, "On the use of stochastic estimator learning automata in time division multiple access systems: A methodology," *Neurocomputing*, vol. 35, pp. 177–188, 2000.

[12] 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, 1996.

[13] X. Zeng, J. Zhou, and C. Vasseur, "A strategy for controlling nonlinear systems using a learning automaton," *Automatica*, vol. 36, pp. 1517–1524, 2000.

[14] J. K. Lanctot and B. J. Oommen, "Discretized estimator learning automata," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 1473–1483, 1992.

[15] A. V. Vasilakos and G. I. Papadimitriou, "A new approach to the design of reinforcement schemes for learning automata: Stochastic estimator learning algorithm," *Neurocomputing*, vol. 7, pp. 275–297, 1995.

[16] B. J. Oommen and T. D. Roberts, "A fast and efficient solution to the capacity assignment problem using discretized learning automata," in *Proc. 11th Int. Conf. Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, vol. II, 1998, pp. 56–65.

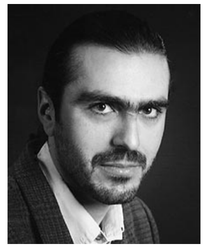
[17] B. J. Oommen and E. R. Hansen, "The asymptotic optimality of discretized linear reward–inaction learning automata," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, pp. 542–545, 1984.

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

[19] B. J. Oommen and J. P. R. Christensen, "ϵ-optimal discretized linear reward–penalty learning automata," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 451–458, 1988.

[20] B. J. Oommen and M. A. Agache, "A comparison of continuous and discretized pursuit learning schemes," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. IV, 1999, pp. 1061–1067.

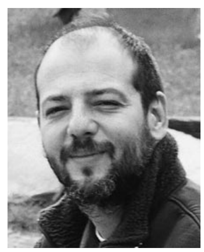
[21] ——, "Continuous and discretized pursuit learning schemes: Various algorithms and their comparison," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, pp. 277–287, Apr. 2001.



Anastasios A. Economides was born in Thessaloniki, Greece. He received the Dipl.Ing. degree in electrical engineering from Aristotle University of Thessaloniki in 1984, and the M.Sc. and Ph.D. degrees in computer engineering from the University of Southern California, Los Angeles, in 1987 and 1990, respectively.

He is currently an Assistant Professor of Computer Networks and Vice-Chairman in the Information Systems Postgraduate Program at the University of Macedonia, Thessaloniki. He is the Director of CONTA (COmputer Networks and Telematics Applications) Laboratory and Coordinator of several projects on tele-education, educational technology, and training trainers. His research interests are in the area of high-speed multimedia networks, tele-education, e-commerce, and learning automata applications.

Dr. Economides received a Fulbright and a Greek State Fellowship.



Athanasios Kehagias was born in Thessaloniki, Greece, in 1961. He received the Dipl.Ing. degree in electrical engineering from Aristotle University of Thessaloniki in 1984, the M.Sc. degree in applied mathematics from Lehigh University, Bethlehem, PA, in 1986, and the Ph.D. degree in applied mathematics from Brown University, Providence, RI, in 1991.

Since November 1999, he has been with the Department of Mathematics, Physical and Computational Sciences, School of Engineering, Aristotle University of Thessaloniki. His research interests include applications of probability theory, algebra, and fuzzy sets.