A. Economides and Ath. Kehagias.
"The STAR Automaton: Expediency and Optimality Properties".

This paper has appeared in the journal:
IEEE Trans. on Systems, Man and Cybernetics, vol.32, pp.723-737, 2002.

The STAR Automaton: Expediency and Optimality Properties

A. Economides
Dept. of Informatics,University of Macedonia
540 06 Thessaloniki, Greece
economid@macedonia.uom.gr

and

Ath. Kehagias
Division of Mathematics
Department of Mathematics, Physical and Computational Sciences
Faculty of Engineering
Aristotle University of Thessaloniki
540 06 Thessaloniki, Greece
e-mail: kehagias@egnatia.ee.auth.gr

June 7, 2001

Abstract

We present the *STack ARchitecture* (STAR) automaton, a fixed structure, multi-action, reward-penalty learning automaton. STAR is characterized by the star shaped structure of its state transition diagram: each branch of the star consists of D states, which are all associated with the same action. There are r branches, one for each action in the action set. An additional "neutral" state also exists, from which each action is selected equiprobably. The learning behavior of STAR results from the stack-like operation of each branch. The learning parameter is the depth D. For $D = 1$, STAR has the same limiting behavior as the L_{R-P} variable structure automaton; for large D, STAR behaves like the $L_{R-\epsilon P}$ variable structure automaton and is ϵ-optimal. STAR has a faster rate of convergence than variable structure automata as observed by numerical simulation and this results in superior behavior in switching environments.

1 Introduction

Early work on learning developed in the context of mathematical psychology [1, 2, 4]. 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 to quantify. The theory of *learning automata* [3] 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, ..., 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)*.

Historically, the theory of learning automata was initiated with the study of FSSA (see [8]). Later, interest shifted to the study of VSSA (see [9]) which appeared to be more adaptable. Currently learning automata research is concentrated mainly on VSSA. An excellent overview is provided in [3].

While VSSA's have attracted a lot of attention, FSSA's are easier to implement and require less computation per time step. This motivated us to explore FSSA designs which perform as well or better than corresponding VSSA's (e.g. are expedient, converge quickly etc.). Good performance combined with simplicity of implementation would make such FSSA's attractive competitors to the currently used VSSA's. Here we present a FSSA with such properties. This is the *STar ARchitecture* (STAR) automaton. The name refers to the star shaped structure of the transition diagram, displayed in Fig.1a. 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}. For the case of large D, we prove that STAR behaves like the $L_{R-\epsilon P}$ automaton and is ϵ- optimal. In general, the D parameter can be fine-tuned to provide the best tradeoff between optimality and speed of response to switching environments. An essential feature of STAR is the possibility for nondeterministic effects of either reward or penalty from the environment. This is controlled via two more parameters δ and ϵ. We present computer simulations which indicate that STAR$^{(D)}$ can outperform VSSA's such as L_{R-P} 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. This idea has been introduced by Oommen [5], in the VSSA context. In [6, 7] action probabilities are updated by the usual VSSA rules; however only a large but *finite* number of discretized probability values is used. As pointed in [6, 7], it is difficult to show ϵ- optimality for the multi-action discrete VSSA's. On the contrary ϵ- optimality of STAR is proven in exactly the same way for the two-action and multi-action case. The proof is rather easy, because of the fact that we use only three values for the action probabilities. Hence, we are able to analyze STAR behavior using relatively simple mathematical tools, such as the theory of finite Markov chains. On the contrary the analysis of variable structure automata requires the use of the theory of stochastic difference equations and more delicate arguments. Especially for the case of $L_{R-\epsilon P}$, analysis is only possible by an approximation argument (see pp.166-168, [3]).

The rest of the paper is organized as follows. In Section 2 we review the fundamental concepts of stochastic learning automata. In Section 3 we present STAR with depth D= 1 and prove its optimality properties. In Section 4 we present STAR with depth $D > 1$ and prove its optimality properties. In Section 5 we present computer simulations to compare the performance of STAR to that of L_{R-P} and $L_{R-\epsilon P}$. Finally in Section 6 we summarize, present our conclusions and propose some directions for future research.

2 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.

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

(i) $\alpha = \{1, 2, ..., r\}$ is the set of actions (input to the environment),

(ii) $\beta = \{0, 1\}$ is the set of responses (output of the environment), and

(iii) $c = \{c_1, c_2, ..., c_r\}$ is an unknown penalty probability set.

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

(i) $\Phi = \{1, 2, ..., s\}$ is the set of the internal states,

(ii) $\alpha = \{1, 2, ..., r\}$ is the set of actions (output of the automaton),

(iii) $\beta = \{0, 1\}$ is the set of responses (input to the automaton),

(iv) $\mathbf{F}(.,.,.) : \Phi * \alpha * \beta \mapsto \Phi$ is the state transition mechanism according to which the next state is chosen (depending on the current state and the environment response).

(v) $\mathbf{G}(.) : \Phi \mapsto \alpha$ 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 probabilistically (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 $p_i(n) = Prob[a(n) = i]$; 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: $c_i = Prob[\beta(n) = 1|a(n) = i]$ $\forall i$. Thus, the environment is characterized by the set of penalty probabilities $c = \{c_1, ..., c_r\}$. The environment reward probability is $d_i = 1 - c_i$, $i = 1, ..., r$. It should be emphasized that the environment penalty probabilities $\{c_i\}$ are unknown to the automaton.

It is desired that the automaton selects the action that will produce the minimum environment penalty probability $c^* = \min_i\{c_i\}$. Typically, the performance is measured by the average cost for a given action probability vector $M(n) = E[\beta(n)|p(n)] = Prob[\beta(n) = 1|p(n)] = \sum_{i=1}^{r} Prob[\beta(n) = 1|\alpha(n) = i]p_i(n) = \sum_{i=1}^{r} c_i p_i(n)$. Thus, the action a^* producing the c^* is the best action. Recall that the penalty probabilities are unknown to the automaton. With no a priori information, the automaton selects the actions with equal probability $p_i(n) = \frac{1}{r}$, $i = 1, ..., r$. This is called a pure-chance automaton. Then the average cost is the mean of the penalty probabilities $M_0 = \frac{1}{r}\sum_{i=1}^{r} c_i$.

An automaton that collects information about the environment may perform better than the pure chance automaton. This, a form of learning, takes 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 the 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 $\mathbf{T}$: $p(n+1) = \mathbf{T}(p(n), \alpha(n), \beta(n))$. In other words, a *Learning Automaton* consists of the environment, the automaton and a learning algorithm connecting the environment and the automaton in a feedback configuration.

The basic 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.

Although optimality may appear to be desirable, in nonstationary environments a suboptimal performance may be preferrable, since we do not want the automaton to be locked onto a specific action. Suboptimal automata are described as ϵ-optimal in which the degree of suboptimality may be varied by suitable choice of the updating algorithm. An ϵ-optimal automaton satisfies $\lim_{n\to\infty} E[M(n)] < c^* + \epsilon$, where $\epsilon > 0$.

Up to this point we have not specified what is the state of the automaton, the state transition mechanism and the action selection mechanism. These elements determine the structure of the automaton. The theory of Learning Automata was inaugurated with the study of automata with an abstract, finite state space and fixed transition and action probabilities. Such automata are known as *Fixed Structure Stochastic Automata* (FSSA) [8]. Later, attention shifted to automata where the state *is* the action probability vector, which can take a continuous infinity of values. Such automata are known as *Variable Structure Stochastic Automata* (VSSA) [3]. Classic examples of VSSA are the L_{R-P} and $L_{R-\epsilon P}$. For details see [3].

Despite the extensive attention recently paid to VSSA, we believe that FSSA can perform comparably or better. In this paper we introduce STAR, a new FSSA, and show that it is superior to the L_{R-P} and $L_{R-\epsilon P}$. The details of the STAR structure will be presented in the following two sections.

3 STAR$^{(1)}$

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

As usual, the action set is $\alpha = \{1, ..., r\}$, the environment response set is $\beta = \{0, 1\}$ (reward and penalty) the penalty probabilities are $c_1, ..., c_r$ and the reward probabilities are $d_1, ..., d_r$. The automaton can be in any of $r+1$ states, $\{0, 1, ..., r\}$. Both the state transition and action selection mechanisms are illustrated in Fig. 1a. The star-shaped structure which gives STAR its name, is clearly illustrated in the figure.

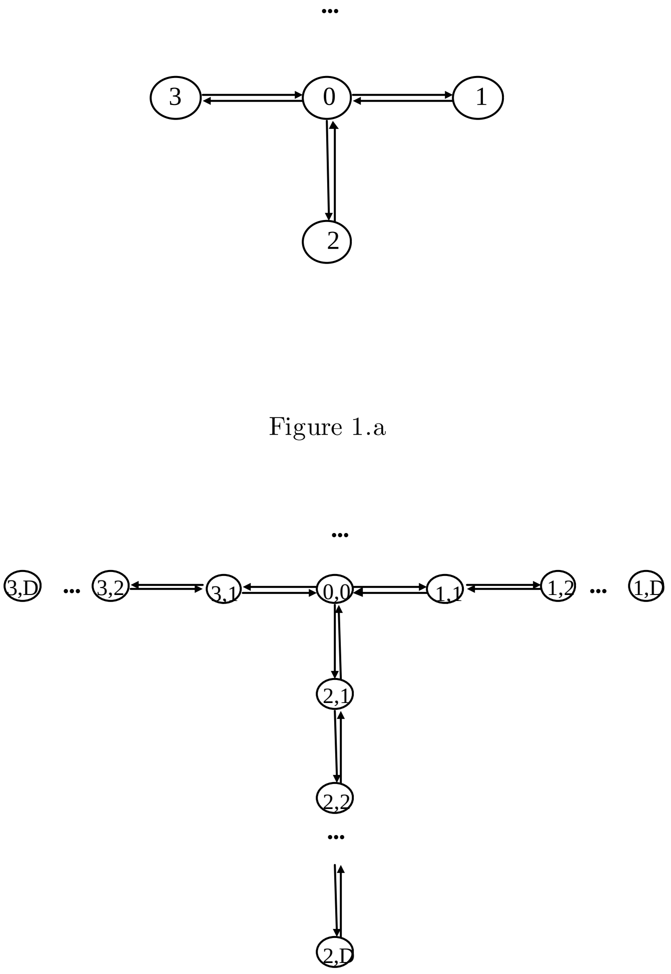

Figure 1.a

Figure 1.b

Figure 1. (a) The structure of STAR$^{(1)}$. (b) The structure of STAR$^{(D)}$.

When the automaton is in state 1, it performs action 1 with probability 1, when in state 2 it performs action 2 with probability 1 and so on for i=1,2,...,r. So each one of these states is “committed” to a corresponding action. On the other hand, the state 0 is a special, so-called “neutral” state: when in that state, the automaton chroooses any of the r actions equiprobably. The action selection mechanism described above can be summarized by the action selection probability $G : \Phi \to \alpha$. We have

$$G_{ij} \doteq Prob[\alpha(n) = j|\Phi(n) = i] = 1 \qquad i = 1, ..., r,\ j = i; \tag{1}$$

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

To evaluate the expediency and optimality of the automaton, we need to know the action probability:

$$p_j(n) \doteq Prob[\alpha(n) = j] \qquad j = 1, ..., r; \tag{3}$$

in vector form, $p(n) \doteq [p_1(n) \; ... \; p_r(n)]$. Now, let us also define the probability of being at state i at time n:

$$\pi_i(n) \doteq Prob[\Phi(n) = i] \qquad i = 0, 1, ..., r; \tag{4}$$

in vector form, $\pi(n) \doteq [\pi_0(n) \; \pi_1(n) \; ... \; \pi_r(n)]$. We have the following relationship between action probabilities and state probabilities

$$p(n) = \pi(n) \cdot G \tag{5}$$

Hence, both the learning behavior and the optimality properties of the automaton depend on the state probabilities $\pi(n)$. However, to say more about these probabilities, we first need to specify the state transition mechanism. This is defined by the probabilities $F : \Phi * \alpha * \beta \rightarrow \Phi$, where

$$F_{ijk,l} \doteq Prob[\Phi(n+1) = l | \Phi(n) = i, \alpha(n) = j, \beta(n) = k] \qquad i, l = 0, 1, ..., r \quad j = 1, ..., r \quad k = 0, 1. \tag{6}$$

These probabilities depend on the current state, action and response, but they are time invariant. In the following paragraphs we will present several possible choices of F. For all of these cases it will turn out that the state process $\Phi(n)$ is a Markov chain. Hence to determine $\pi(n)$ it suffices to study the behavior of the state transition matrix P, with elements

$$P_{il} \doteq Prob[\Phi(n+1) = l | \Phi(n) = i] \qquad i, l = 0, 1, ..., r. \tag{7}$$

In all the cases which we consider, the process $\Phi(n)$ turns out to be ergodic; hence it has a unique equilibrium probability vector $\pi \doteq [\pi_0 \; \pi_1 \; ... \; \pi_r]$, where π_i is defined by

$$\pi_i \doteq \lim_{n \to \infty} \pi_i(n) \qquad i = 0, 1, ..., r. \tag{8}$$

We now proceed to define F. We distinguish four cases.

3.1 Deterministic Reward - Deterministic Penalty

This is the simplest possible case. Both reward and penalty cause deterministic state transitions, according to the following rules.

1. When in state 0 and chosen action is i (i= $1, .., r$), if rewarded go to state i w.p. 1 (with probability one):

$$F_{0i0,i} = 1, \;\; F_{0i0,j} = 0 \qquad i = 1, ..., r, \;\; j = 0, 1, ..., r \;\; j \neq i; \tag{9}$$

   if punished, stay in state 0 w.p. 1:

$$F_{0i1,0} = 1, \;\; F_{0i1,j} = 0 \qquad i, j = 1, ..., r. \tag{10}$$

2. When in state i, $i \neq 0$ and chosen action is i (i= $1, .., r$), if rewarded stay in state i w.p. 1:

$$F_{ii0,i} = 1, \;\; F_{ii0,j} = 0, \qquad i = 1, ..., r \;\; j = 0, 1, ..., r \;\; j \neq i; \tag{11}$$

   if punished, go to state 0 w.p. 1:

$$F_{ii1,0} = 1, \;\; F_{ii1,j} = 0, \qquad i, j = 1, ..., r. \tag{12}$$

Having defined F we can compute the equilibrium state probabilities π and action probabilities p; from these we can infer the expediency of the automaton. For the sake of brevity, we only present the main results of our analysis. Detailed derivations can be found in the Appendix.

From F we find that the nonzero elements of P are

$$P_{00} = \frac{1}{r} \sum_{i=1}^{r} c_i, \quad P_{0i} = \frac{1 - c_i}{r}, \quad P_{i0} = c_i, \quad P_{ii} = 1 - c_i, \tag{13}$$

for $i = 1, ..., r$; all the other elements of P are zero. Now, from eq.(13) it is obvious that $P_{ii} > 0$ for $i = 0, 1, ..., r$. Also, it is easy to check that $P^2 > 0$. Hence, the state process $\Phi(n)$ is irreducible and aperiodic, which means that it is also ergodic (see [3]). It follows that, from P we can compute the state probabilities π, which turn out to be

$$\pi_0 = \frac{r}{\sum_{i=1}^{r} \frac{1}{c_i}}, \quad \pi_i = \frac{r}{\sum_{j=1}^{r} \frac{1}{c_j}} \cdot \frac{1 - c_i}{c_i}, \qquad i = 1, 2, ..., r. \tag{14}$$

Now, taking the limit of (5) as $n \to \infty$ we obtain the limit action probabilities as $p = \pi G$ and finally find

$$p_j = \frac{\frac{1}{c_j}}{\sum_{i=1}^{r} \frac{1}{c_i}} \qquad j = 1, ..., r. \tag{15}$$

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

$$M_1 = \sum_{j=1}^{r} c_j \cdot p_j = \sum_{j=1}^{r} c_j \cdot \frac{\frac{1}{c_j}}{\sum_{i=1}^{r} \frac{1}{c_i}} = \sum_{j=1}^{r} \frac{1}{\sum_{i=1}^{r} \frac{1}{c_i}} = \frac{r}{\sum_{i=1}^{r} \frac{1}{c_i}}. \tag{16}$$

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

3.2 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, .., r), if rewarded go to state i w.p. 1:

$$F_{0i0,i} = 1, \quad F_{0i0,j} = 0, \qquad i = 1, ..., r \;\; j = 0, 1, ..., r \;\; j \neq i; \tag{17}$$

but if punished, go to state i w.p. δ or stay in state 0 w.p. $1 - \delta$:

$$F_{0i1,i} = \delta, \quad F_{0i1,0} = 1 - \delta, \quad F_{0i1,j} = 0 \qquad i, j = 1, ..., r \;\; j \neq i. \tag{18}$$

2. When in state i, $i \neq 0$ and chosen action is i (i= 1, .., r), if rewarded stay in state i w.p. 1:

$$F_{ii0,i} = 1, \quad F_{ii0,j} = 0, \qquad i = 1, ..., r \;\; j = 0, 1, ..., r \;\; j \neq i; \tag{19}$$

if punished, stay in state i w.p. δ, or go to state 0 w.p. $1 - \delta$:

$$F_{ii1,i} = \delta, \quad F_{ii1,0} = 1 - \delta, \quad F_{ii1,j} = 0, \qquad i, j = 1, ..., r, \;\; j \neq i. \tag{20}$$

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

$$P_{00} = \frac{1 - \delta}{r} \cdot \sum_{i=1}^{r} c_i, \quad P_{0i} = \delta \cdot \frac{c_i}{r} + \frac{1 - c_i}{r}, \tag{21}$$

$$P_{i0} = (1 - \delta) \cdot c_i, \quad P_{ii} = 1 - c_i + \delta \cdot c_i, \tag{22}$$

for $i = 1, ..., r$; all the other elements of P are zero. We use the same argument as previously , to show that $\Phi(n)$ is ergodic. Hence from P we compute π and p; π turns out to be of a form similar to that of (15):

$$\pi_0 = \frac{r}{\sum_{i=1}^{r} \frac{1}{\hat{c}_i}}, \quad \pi_i = \frac{r}{\sum_{j=1}^{r} \frac{1}{\hat{c}_j}} \cdot \frac{1-\hat{c}_i}{\hat{c}_i}, \qquad i = 1, 2, ..., r. \tag{23}$$

We observe that (23) is of exactly the same form as (15), except that in place of c_i we have $\hat{c}_i = (1-\delta) \cdot c_i$, $i = 1, ..., r$. Hence, similarly to the previous case, we find

$$p_j = \frac{\frac{1}{\hat{c}_j}}{\sum_{i=1}^{r} \frac{1}{\hat{c}_i}} \qquad j = 1, ..., r \tag{24}$$

and we compute the limiting average cost as

$$M_1^\delta = \frac{r}{\sum_{i=1}^{r} \frac{1}{\hat{c}_i}} = (1-\delta) \cdot \frac{r}{\sum_{i=1}^{r} \frac{1}{c_i}} < M_1. \tag{25}$$

Hence, in the limit, 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. [1]

3.3 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= $1, .., r$), if rewarded go to state i w.p. $1-\epsilon$ or stay in state 0 w.p. ϵ:

$$F_{0i0,i} = 1-\epsilon, \; F_{0i0,0} = \epsilon, \; F_{0i0,j} = 0, \qquad i, j = 1, ..., r \;\; j \neq i; \tag{26}$$

   but if punished, stay in state 0 w.p. 1:

$$F_{0i1,0} = 1, \; F_{0i1,j} = 0 \qquad i, j = 1, ..., r. \tag{27}$$

2. When in state i, $i \neq 0$ and chosen action is i (i= $1, .., r$), if rewarded stay in state i w.p. 1-ϵ or go to state 0 w.p. ϵ:

$$F_{ii0,i} = 1-\epsilon, \; F_{ii0,0} = \epsilon, \; F_{ii0,j} = 0 \qquad i, j = 1, ..., r, \;\; j \neq i; \tag{28}$$

   if punished, go to state 0 w.p. 1:

$$F_{ii1,0} = 1, \quad F_{ii1,j} = 0, \qquad i, j = 1, ..., r. \tag{29}$$

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

$$P_{00} = \frac{1}{r}\sum_{i=1}^{r} c_i + \epsilon \cdot \frac{1}{r}\sum_{i=1}^{r}(1-c_i), \quad P_{0i} = (1-\epsilon) \cdot \frac{1-c_i}{r} \tag{30}$$

[1] From eq.(25) it may appear that in the case $\delta = 1$, M_1^δ=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 $i = 1, ..., r$ 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 eq.(20). 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.

$$P_{i0} = c_i + \epsilon \cdot (1 - c_i), \quad P_{ii} = (1 - \epsilon) \cdot (1 - c_i), \tag{31}$$

for $i = 1, ..., r$; all the other elements of P are zero. From P we infer that $\Phi(n)$ is ergodic and compute π and p; π turns out to be of a form similar to that of (14):

$$\pi_0 = \frac{r}{\sum_{i=1}^{r} \frac{1}{\bar{c}_i}}, \quad \pi_i = \frac{r}{\sum_{j=1}^{r} \frac{1}{\bar{c}_j}} \cdot \frac{1 - \bar{c}_i}{\bar{c}_i}, \qquad i = 1, 2, ..., r. \tag{32}$$

We observe that this of exactly the same form as (15), except that in place of c_i we have $\bar{c}_i = c_i + \epsilon \cdot (1 - c_i) > c_i$, $i = 1, ..., r$. Similarly to the previous case, we find

$$p_j = \frac{\frac{1}{\bar{c}_j}}{\sum_{i=1}^{r} \frac{1}{\bar{c}_i}} \qquad j = 1,...,r \tag{33}$$

and compute the limiting average cost to be

$$M_1^\epsilon = \frac{r}{\sum_{i=1}^{r} \frac{1}{\bar{c}_i}}. \tag{34}$$

Using $\bar{c}_i > c_i$, $i = 1,...,r$, it is easy to prove that $M_1^\epsilon > M_1$, which shows that STAR$^{(1)}$ with probabilistic reward and deterministic penalty will perform worse than STAR$^{(1)}$ with deterministic reward and deterministic penalty, as well as the L_{R-P}.

3.4 Probabilistic Reward - Probabilistic Penalty

This is the most general case: both reward and penalty cause probabilistic state transitions, according to the following rules.

1. When in state 0 and chosen action is i (i= 1,..,r), if rewarded go to state i w.p. $1 - \epsilon$ or stay in state 0 w.p. ϵ:

$$F_{0i0,i} = 1 - \epsilon, \;\; F_{0i0,0} = \epsilon, \;\; F_{0i0,j} = 0, \qquad i, j = 1,...,r \;\; j \neq i; \tag{35}$$

but if punished, go to state i w.p. δ, or stay in state 0 w.p. $1 - \delta$:

$$F_{0i1,i} = \delta, \;\; F_{0i1,0} = 1 - \delta, \;\; F_{0i1,j} = 0 \qquad i, j = 1, ..., r \;\; j \neq i. \tag{36}$$

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

$$F_{ii0,i} = 1 - \epsilon, \;\; F_{ii0,0} = \epsilon, \;\; F_{ii0,j} = 0 \qquad i, j = 1, ..., r \;\; j \neq i; \tag{37}$$

if punished, go to state 0 w.p. $1 - \delta$, or stay in state i w.p. δ:

$$F_{ii1,i} = \delta, \;\; F_{ii1,0} = 1 - \delta, \;\; F_{ii1,j} = 0, \qquad i, j = 1, ..., r, \;\; j \neq i. \tag{38}$$

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

$$P_{00} = \frac{1 - \delta}{r} \cdot \sum_{i=1}^{r} c_i + \frac{\epsilon}{r} \cdot \sum_{i=1}^{r} 1 - c_i \quad P_{0i} = \delta \cdot \frac{c_i}{r} + (1 - \epsilon) \cdot \frac{1 - c_i}{r} \tag{39}$$

$$P_{i0} = (1 - \delta) \cdot c_i + \epsilon \cdot (1 - c_i), \quad P_{ii} = \delta \cdot c_i + (1 - \epsilon) \cdot (1 - c_i), \tag{40}$$

for $i = 1, ..., r$; all the other elements of P are zero. By the same arguments as previously it is seen that $\Phi(n)$ is ergodic; its equilibrium state probabilities π turn out similar to (15) 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 $\delta = 0$, $0 < \epsilon < 1$.

4 STAR$^{(D)}$

In this section we present the STAR$^{(D)}$ automaton with arbitrary depth D. The presentation is similar to that of the previous section, concerning STAR$^{(1)}$, hence this section will be briefer. Once again, proofs are relegated to the Appendix.

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 labelling 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, j) is the j-th state committed to action i. Hence the index i runs from 1 to r and the index j runs from 1 to D.

This numbering of the states corresponds to the star-shaped structure of Fig.1b. The idea is the following: states are partitioned into r sets of D states each, each such set forming a *branch* of the star, the particular 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 i-th branch; when it is punished it moves towards 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 $G : \Phi \rightarrow \alpha$ (but note that the state set Φ is different from that of STAR$^{(1)}$!).

$$G_{(i,d),j} \doteq Prob[\alpha(n) = j|\Phi(n) = (i, d)] = 1 \qquad i, j = 1, ..., r, \;\; d = 1, ..., D, \tag{41}$$

$$G_{(0,0),j} \doteq Prob[\alpha(n) = j|\Phi(n) = (0, 0)] = \frac{1}{r} \qquad j = 1, ..., r. \tag{42}$$

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

$$p_j(n) \doteq Prob[\alpha(n) = j] \qquad j = 1, ..., r; \tag{43}$$

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

$$\pi_{(i,d)}(n) \doteq Prob[\Phi(n) = (i, d)] \qquad (i, d) = (0, 0) \text{ or } \;\; i = 1, ..., r, \;\; d = 1, ..., D; \tag{44}$$

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

$$p_j(n) = \sum_{\forall(i,d)} \pi_{(i,d)}(n) \cdot G_{(i,d),j}. \tag{45}$$

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

$$\pi_{(i,d)} \doteq \lim_{n\rightarrow\infty} \pi_{(i,d)}(n), \qquad (i, d) = (0, 0) \text{ or } \;\; i = 1, ..., r, \;\; d = 1, ..., D. \tag{46}$$

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

$$F_{(i,d)jk,(i',d')} \doteq Prob[\Phi(n + 1) = (i', d')|\Phi(n) = (i, d), \alpha(n) = j, \beta(n) = k]. \tag{47}$$

We first present the most general F we use. This is characterized by probabilistic reward and probabilistic penalty, and corresponds to the STAR$^{(1)}$ case of Section 3.4; that is, $0 < \delta < 1$ and $0 < \epsilon < 1$. Then, we will consider some special cases of F, which yield convenient expressions for the state and action probabilities. 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 $(i,1)$ w.p. $1-\epsilon$ or stay in state $(0,0)$ w.p. ϵ,

$$F_{(0,0)i0,(i,1)} = 1-\epsilon, \quad F_{(0,0)i0,(0,0)} = \epsilon \qquad i=1,...,r; \tag{48}$$

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

$$F_{(0,0)i1,(i,1)} = \delta, \quad F_{(0,0)i1,(0,0)} = 1-\delta \qquad i=1,...,r. \tag{49}$$

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

$$F_{(i,1)i0,(i,2)} = 1-\epsilon, \quad F_{(i,1)i0,(0,0)} = \epsilon \qquad i=1,...,r; \tag{50}$$

   but, if punished, go to state $(i,2)$ w.p. δ or go to state $(0,0)$ w.p. $1-\delta$,

$$F_{(i,1)i1,(i,2)} = \delta, \quad F_{(i,1)i1,(0,0)} = 1-\delta \qquad i=1,...,r. \tag{51}$$

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

$$F_{(i,d)i0,(i,d+1)} = 1-\epsilon, \quad F_{(i,d)i0,(i,d-1)} = \epsilon \qquad i=1,...,r; \tag{52}$$

   but, if punished, go to state $(i,d+1)$ w.p. δ or go to state $(i,d-1)$ w.p. $1-\delta$,

$$F_{(i,d)i1,(i,d+1)} = \delta, \quad F_{(i,d)i1,(i,d-1)} = 1-\delta \qquad i=1,...,r. \tag{53}$$

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

$$F_{(i,D)i0,(i,D)} = 1-\epsilon, \quad F_{(i,D)i0,(i,D-1)} = \epsilon \qquad i=1,...,r; \tag{54}$$

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

$$F_{(i,D)i1,(i,D)} = \delta, \quad F_{(i,D)i1,(i,D-1)} = 1-\delta \qquad i=1,...,r. \tag{55}$$

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$^{(D)}$. Details of the derivation are presented in the Appendix; the final result is that

$$\begin{array}{lll}
P^{(D)}_{(0,0),(0,0)} = \dfrac{1}{r}\displaystyle\sum_{i=1}^{r}\left[(1-c_i)\cdot\epsilon + c_i\cdot(1-\delta)\right] & P^{(D)}_{(0,0),(i,1)} = \dfrac{1}{r}\left[(1-c_i)\cdot(1-\epsilon)+c_i\cdot\delta\right] & i=1,...,r. \\[2ex]
P^{(D)}_{(i,1),(i,2)} = (1-c_i)\cdot(1-\epsilon)+c_i\cdot\delta & P^{(D)}_{(i,1),(0,0)} = (1-c_i)\cdot\epsilon + c_i\cdot(1-\delta) & i=1,...,r. \\[2ex]
P^{(D)}_{(i,d),(i,d+1)} = (1-c_i)\cdot(1-\epsilon)+c_i\cdot\delta & P^{(D)}_{(i,d),(i,d-1)} = (1-c_i)\cdot\epsilon + c_i\cdot(1-\delta) & i=1,...,r, \\
 & & d=2,...,D-1. \\
P^{(D)}_{(i,D),(i,D)} = (1-c_i)\cdot(1-\epsilon)+c_i\cdot\delta & P^{(D)}_{(i,D),(i,D-1)} = (1-c_i)\cdot\epsilon + c_i\cdot(1-\delta) & i=1,...,r.
\end{array}$$

All the transition probabilities not indicated above are equal to zero. A tedious but straightforward computation shows that $\left(P^{(D)}\right)^{2D} > 0$. Intuitively, this corresponds to the fact that in $2D$ steps,we can get from any state to any other state, with positive probability. Also, 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 $\pi^{(D)}(n+1) = \pi^{(D)}(n)P^{(D)}$, $\pi^{(D)} = \pi^{(D)}P$, $\pi^{(D)} = \lim_{\to\infty}\pi^{(D)}(n)$. Just like in the STAR$^{(1)}$ case, from $\pi^{(D)}$ we obtain a limiting (equilibrium) action probability vector $p^{(D)} = [p_1^{(D)}...p_r^{(D)}]$.

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

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

$$p_i^{(D)} = \frac{\sum_{d=0}^{D} (\frac{1-c_i}{c_i})^d}{\sum_{j=1}^{r}\sum_{d=0}^{D} (\frac{1-c_j}{c_j})^d} \qquad i = 1, 2, ..., r. \tag{56}$$

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

$$p_i^{(D)} = \frac{\sum_{d=0}^{D} (\frac{1-\hat{c}_i}{\hat{c}_i})^d}{\sum_{j=1}^{r}\sum_{d=0}^{D} (\frac{1-\hat{c}_j}{\hat{c}_j})^d} \qquad i = 1, 2, ..., r. \tag{57}$$

   This is exactly of the same form as (56), except that, in place of c_i we use $\hat{c}_i = (1 - \delta) \cdot c_i$.

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

$$p_i^{(D)} = \frac{\sum_{d=0}^{D} (\frac{1-\bar{c}_i}{\bar{c}_i})^d}{\sum_{j=1}^{r}\sum_{d=0}^{D} (\frac{1-\bar{c}_j}{\bar{c}_j})^d} \qquad i = 1, 2, ..., r. \tag{58}$$

   This is exactly of the same form as (56), except that, in place of c_i we use $\bar{c}_i = c_i + \epsilon \cdot (1 - c_i)$.

Note that in each of the above cases, when $D = 1$, we recover $p^{(1)} = p$ as given in the previous section, in other words we recover the STAR$^{(1)}$ case.

From equation (56) we obtain ϵ- optimality of the STAR$^{(D)}$ automaton, provided there is at least one $c_i < 0.5$. To show this, without loss of generality, assume that c_1 is the minimum of $\{c_1, \dots, c_r\}$ and that $c_1 < 0.5$. Define $g_i = (1 - c_i)/c_i$ for $i = 1, ..., r$ and note that g_1 is the maximum of $\{g_1, ..., g_r\}$ and, in fact, $g_1 > 1$. To show ϵ-optimality, we only need to show that $p_1^{(D)} \to 1$ as $D \to \infty$. To show this, choose any i= 2, ..., r and take the ratio

$$\frac{p_i^{(D)}}{p_1^{(D)}} = \frac{\sum_{d=0}^{D} (\frac{1-c_i}{c_i})^d}{\sum_{d=0}^{D} (\frac{1-c_1}{c_1})^d} = \frac{1 + g_i + ... + g_i^D}{1 + g_1 + ... + g_1^D} = \frac{g_1 - 1}{g_i - 1} \cdot \frac{g_i^{D+1} - 1}{g_1^{D+1} - 1}. \tag{59}$$

The first fraction is independent of D and does not affect convergence. As for the second fraction, there are two cases. If $g_i < 1$, then the numerator tends to -1 and the denominator to ∞, hence the fraction goes to zero. If, on the other hand, $g_i > 1$, then the whole fraction tends to $(g_i/g_1)^D$. But we have assumed that $g_i < g_1$, hence $(g_i/g_1)^D$ goes to zero again. Hence we have proven that $p_i^{(D)}/p_1^{(D)}$ goes to zero as D goes to infinity, for $i = 2, ..., r$. This and the fact that for every D we have $\sum_{i=1}^{r} p_i^D = 1$, shows 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. A similar analysis holds for the probabilistic reward - deterministic penalty case (provided that there is at least one $\hat{c}_i < 0.5$) and for the deterministic reward - probabilistic penalty case, (provided that there is at least one $\bar{c}_i < 0.5$). Finally, it should be noted that, since $\hat{c}_i = (1-\delta)\cdot c_i$, in the probabilistic reward - deterministic penalty case, using a large δ -value increases the chance that $\hat{c}_i < 0.5$, and consequently the chance that the automaton will be ϵ - optimal. However, the value $\delta = 1$ must be avoided because it leads to non-ergodic $\Phi(n)$, for the same reasons discussed in the STAR$^{(1)}$ case.

Note that STAR$^{(1)}$ is expedient, just like L_{R-P}, and STAR$^{(D)}$ is ϵ-optimal, just like $L_{R-\epsilon P}$. However as will be seen in the next section, the STAR automata (both for $D=1$ and $D>1$) are 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. We have computed exactly their steady state action probabilities (except for the case of $\delta > 0$ and $\epsilon > 0$) using the theory of finite Markov chains. In contrast, computation of the steady state probabilities for the $L_{R-\epsilon P}$ automaton requires the use of stochastic difference equations and an approximation argument (see pp.166-168, [3]).

5 Experiments

In this section we present some computer simulation results to corroborate our theoretical analysis. We compare the performance of STAR$^{(D)}$ (for various values of depth D) to that of L_{R-P} and $L_{R-\epsilon P}$ (for various values of learning rates a and b). The automata operate in a switching environment where the best action changes periodically. The automata are compared for various values of the environment penalty probabilities.

An experiment is determined by several automaton and environment parameters. The automata parameters are: depth D for STAR$^{(D)}$, learning rate a for L_{R-P}, and learning rates a (reward) and b (punishment) for $L_{R-\epsilon P}$. The environment parameters are as follows. We have always used ten different actions. Hence there are ten different penalty probabilities c_i, $i = 1,2,...,10$. We always choose action 1 to have the smallest penalty probability: $c_1 < c_i$, $i = 2,...,10$; but as already mentioned the environment is not stationary. Namely, every 50 time steps the values c_1 and c_2 are exchanged, making alternately action 1 or action 2 the best action. Hence, in every period of 50 time steps, the learning automaton has to adjust anew to the best action; we emphasize that the automaton does not know the c_i values. 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 chosen because it was expected that steady-state behavior is reached by the end of the experiment [2]. This assumption was justified experimentally: simulations with longer run lengths, e.g. 10000 steps, showed no appreciable change of the average cost. In fact, convergence of average cost occurs within at most 2000 steps in every experiment.

Each 5000-steps run is repeated ten times and the average costs incurred by the STAR$^{(D)}$, L_{R-P} 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 $M_{opt} = c^*$, is as explained in Section 2, and M_{th} is given by eq.(16).

Hence, for all experiments we use identical environment parameters, except for the choice of c_i, $i = 1,..,10$. This choice determines the particular experiment; we have chosen nine different sets of c_i's resulting in nine different experiments. For every experiment we have tried ten different values of depth D: 1, 2, ..., 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, the punishment learning rate b must also be determined; two choices have been used: $b = a/10$ and $b = a/5$. Our c_i choices are summarized in Table 1.

In Figs. 2, 3, 4, we compare average cost for all D, a and b values, for three representative experiments. We see that in every experiment the best STAR$^{(D)}$ outperforms the best L_{R-P} and $L_{R-\epsilon P}$; and in general most STAR$^{(D)}$'s outperform most L_{R-P} and $L_{R-\epsilon P}$.

[2] The term "steady-state" requires explanation. As the environment penalty probabilities are periodically switched, also the action probabilities change values with the same period, as the automata attempt to follow the environement evolution. Hence the action probabilities never reach steady-state. However, after the first couple thousand steps the time averaged cost incurred by each automaton reaches equilibrium and fluctuates around a mean value. This is true of both the fixed and variable structure automata used in the experiments.

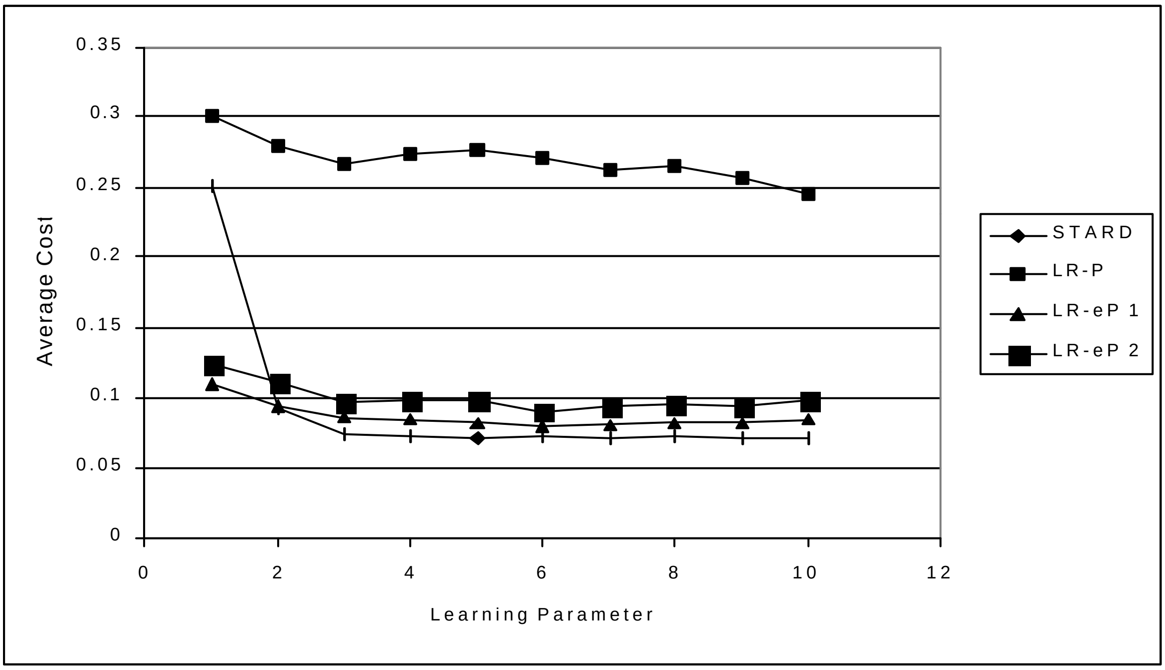

Figure 2. Experiment nr.3. Average STAR$^{(D)}$ cost as a function of D, average L_{R-P} cost as a function of a, average $L_{R-\epsilon P}$ cost as a function of a, b.

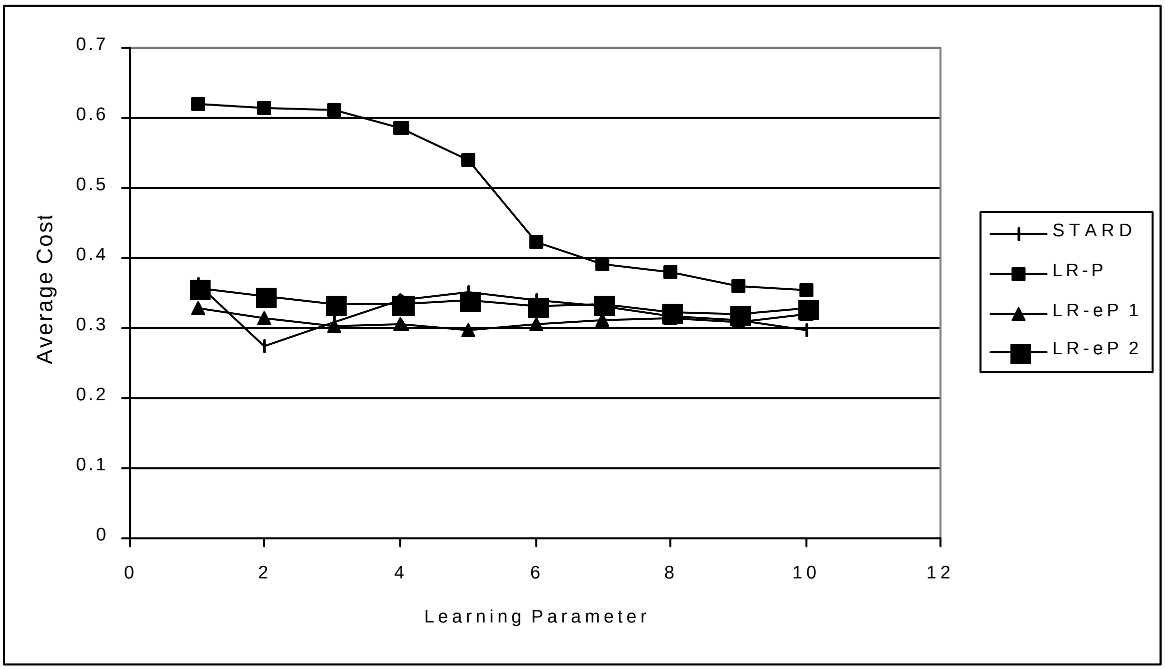

Figure 3. Experiment nr.4. Average STAR$^{(D)}$ cost as a function of D, average L_{R-P} cost as a function of a, average $L_{R-\epsilon P}$ cost as a function of a, b.

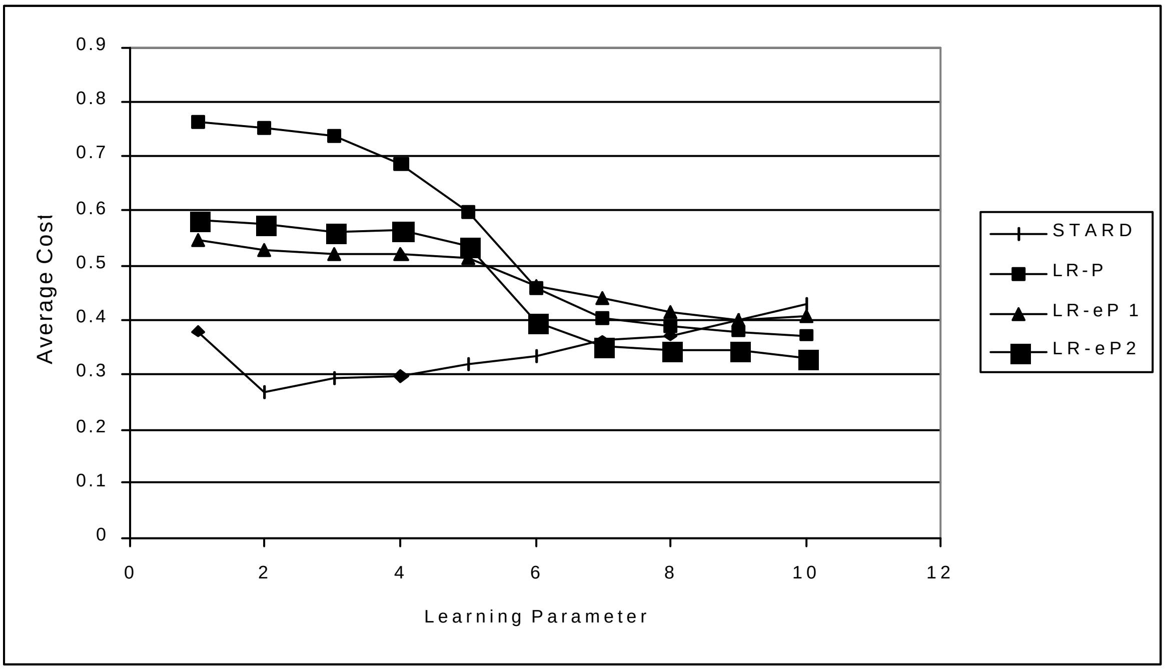

Figure 4. Experiment nr.6. Average STAR$^{(D)}$ cost as a function of D, average L_{R-P} cost as a function of a, average $L_{R-\epsilon P}$ cost as a function of a, b.

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
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

Table 1. Experiment Parameters

Similar patterns occur in the remaining six experiments, as can be seen in Table 1, where we present our cumulative results. Namely, we compare the cost of the best STAR$^{(D)}$ to that of the best L_{R-P} and $L_{R-\epsilon P}$, as well as to theoretical cost M_{th} and optimal cost M_{opt}; finally we also list the cost of STAR$^{(2)}$. 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 $L_{R-\epsilon P}$, while in experiments 8 and 9 STAR$^{(D)}$ and $L_{R-\epsilon P}$ performance is very close. This can be seen in Fig.5 (referring to experiment nr.8) and in Fig.6 (referring to experiment nr.9). In particular, for experiment nr.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 0.890 and 0.897 (note the reduced scaling of the y-axis in Fig. 6).

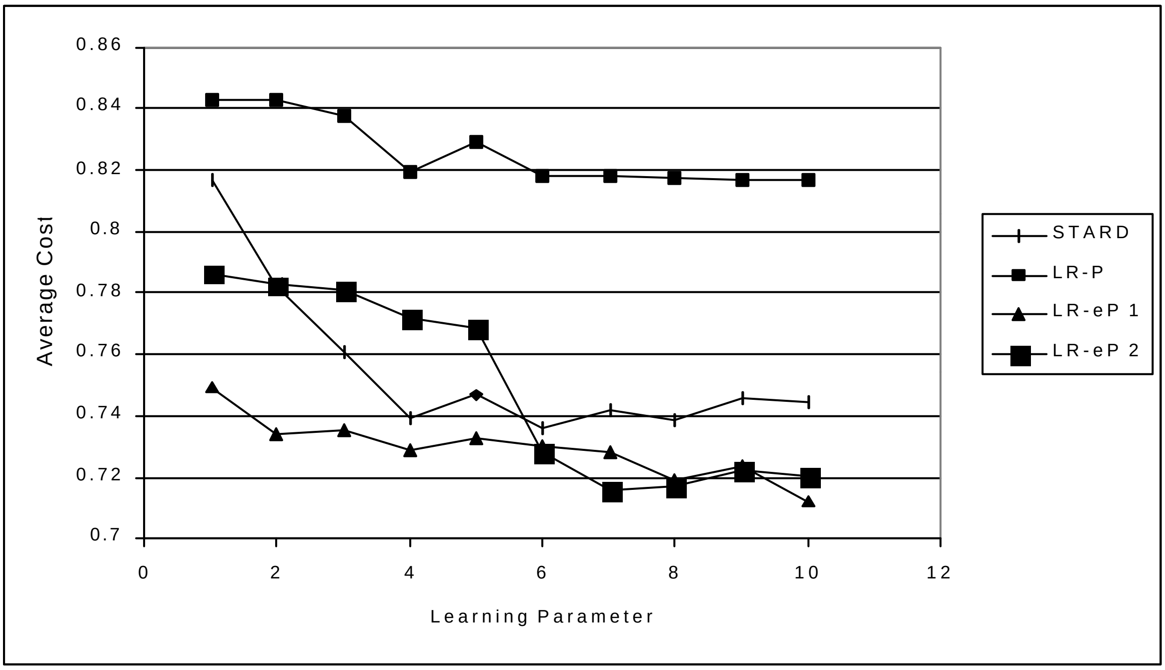

Figure 5. Experiment nr.8. Average STAR$^{(D)}$ cost as a function of D, average L_{R-P} cost as a function of a, average $L_{R-\epsilon P}$ cost as a function of a, b.

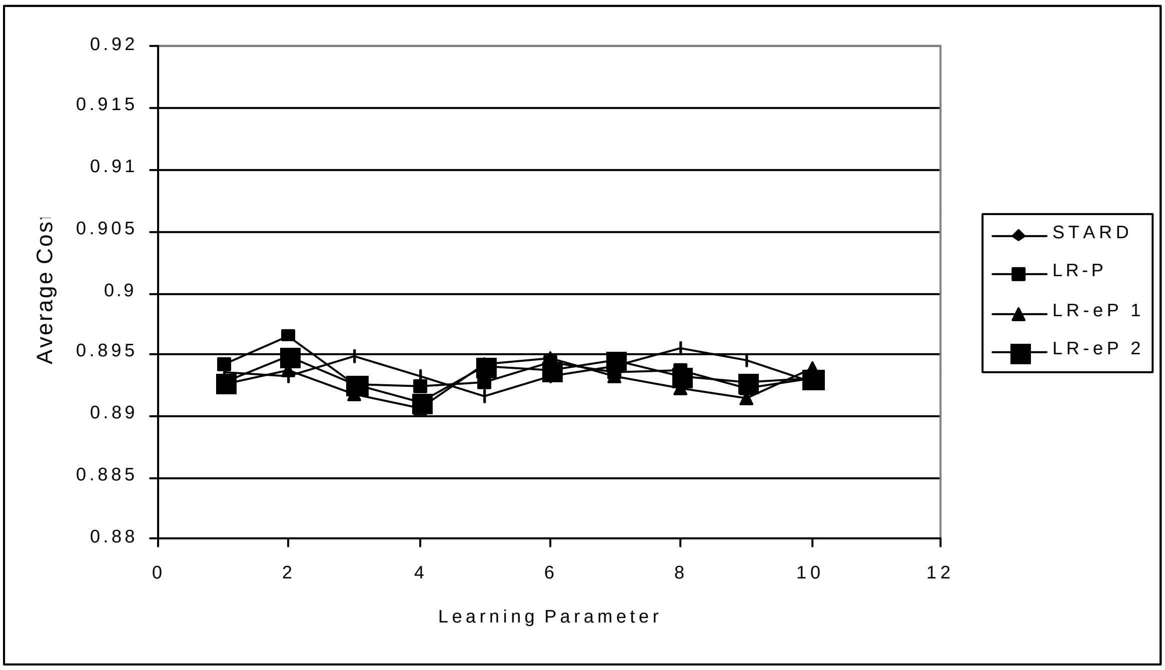

Figure 6. Experiment nr.9. Average STAR$^{(D)}$ cost as a function of D, average L_{R-P} cost as a function of a, average $L_{R-\epsilon P}$ cost as a function of a, b.

In an unknown environment we will not know in advance what is the optimal value of D, a or b to use. However, in Table 1, we also see that the value $D = 2$ yields uniformly good results for all environments tested. We also see that 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 Figs.7a-7c (referring to experiment nr. 4) and 8a-8c (referring to experiment nr. 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.7a 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 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 succesfully following the environment. L_{R-P} with $a = 0.99$ has a much slower response and is less succesful 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 a FSS automaton.

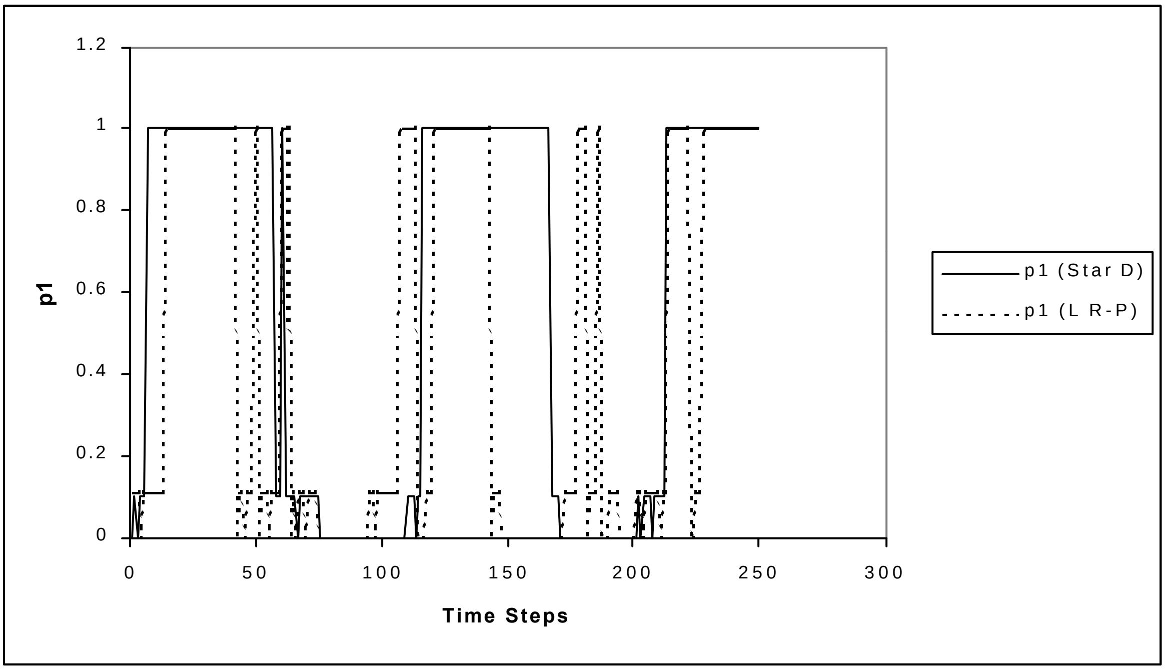

Figure 7a. Experiment nr.4. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and L_{R-P} $(a = 0.99)$ p_1 probability (dotted line).

In Fig.7b we compare action probabilities p_1 of STAR$^{(2)}$ and $L_{R-\epsilon P}$ with $a = 0.20$, $b = 0.02$. Since the STAR p_1 is the same as in the previous figure, 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.

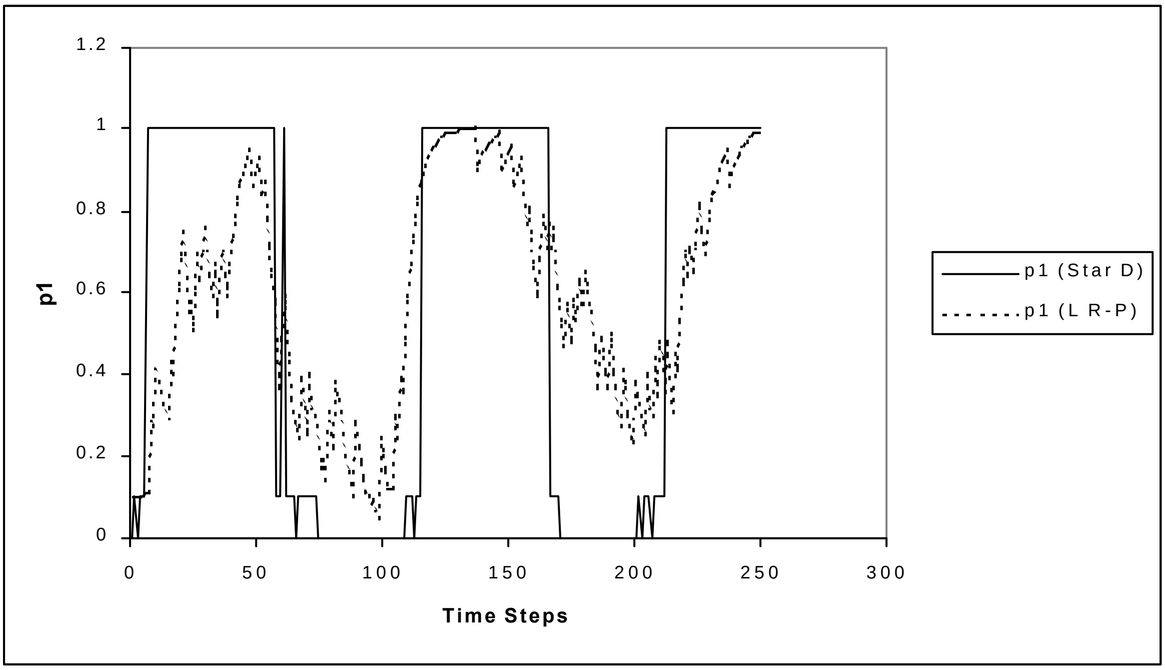

Figure 7b. Experiment nr.4. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and $L_{R-\epsilon P}$ $(a = 0.20, b = 0.02)$ p_1 probability (dotted line).

In Fig.7c we compare action probabilities p_1 of STAR$^{(2)}$ and $L_{R-\epsilon P}$ with $a = 0.90$, $b = 0.18$. Again, $L_{R-\epsilon P}$ is slower and less succesful than STAR in responding to environment switchings. The high a, b values again result in the $L_{R-\epsilon P}$ displaying near-FSSA behavior.

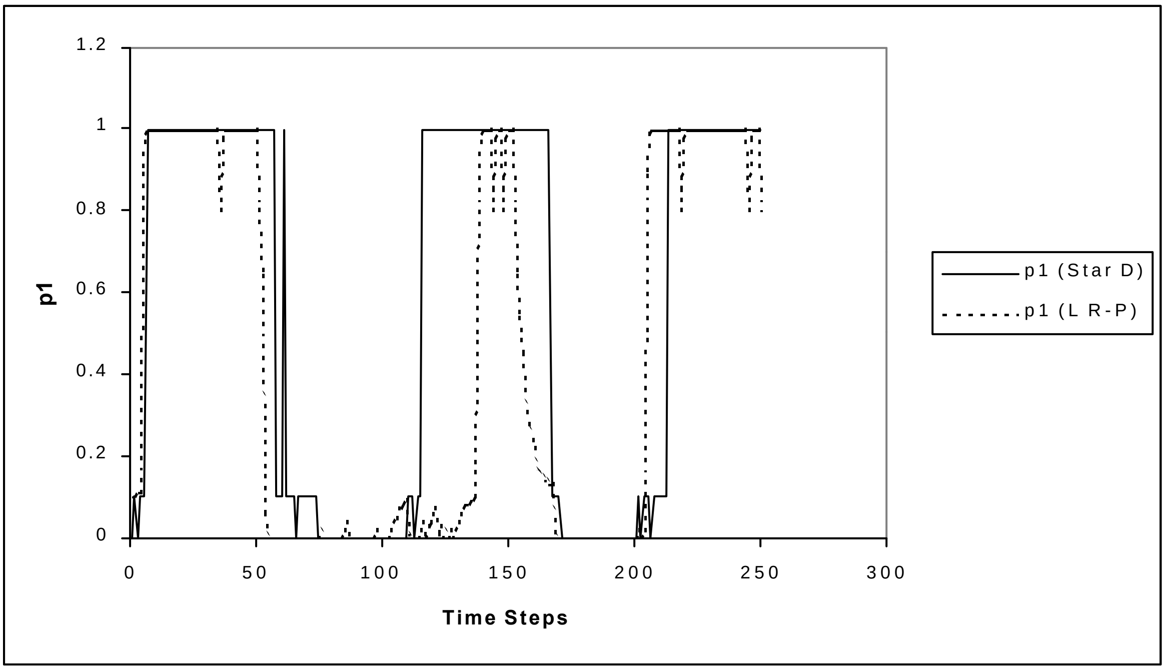

Figure 7c. Experiment nr.4. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and $L_{R-\epsilon P}$ $(a = 0.90, b = 0.18)$ p_1 probability (dotted line).

Similar conclusions can be drawn from Figs.8a- 8c, which pertain to experiment nr.6. Moreover STAR responds to environment switchings almost instantaneously.

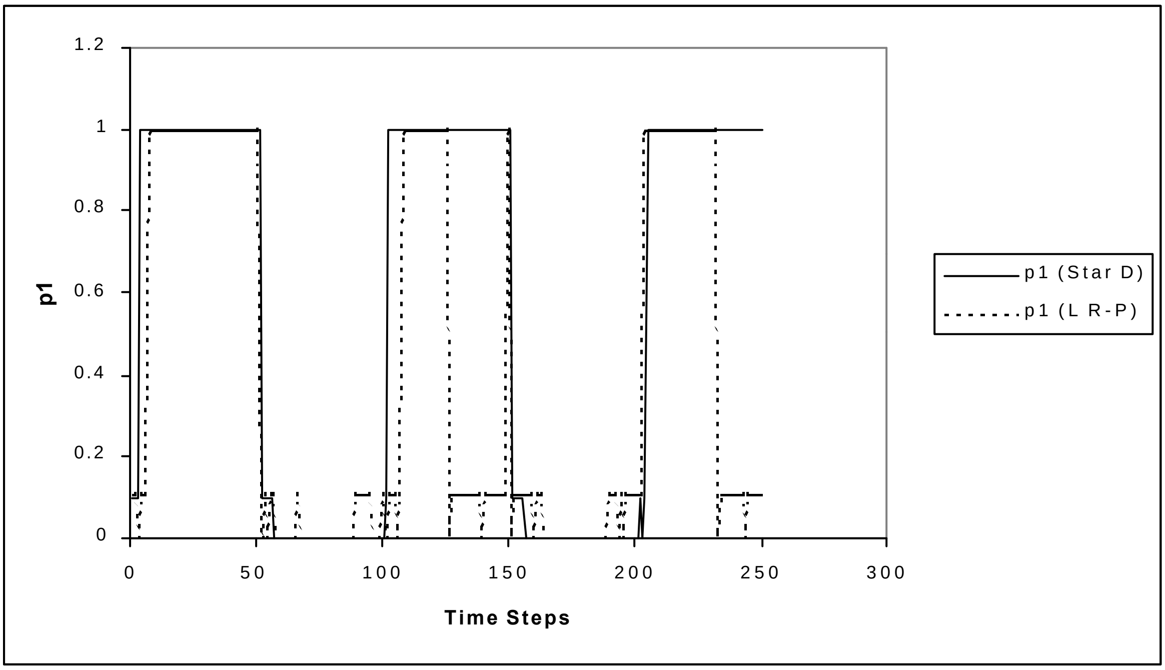

Figure 8a. Experiment nr.6. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and L_{R-P} $(a = 0.99)$ p_1 probability (dotted line).

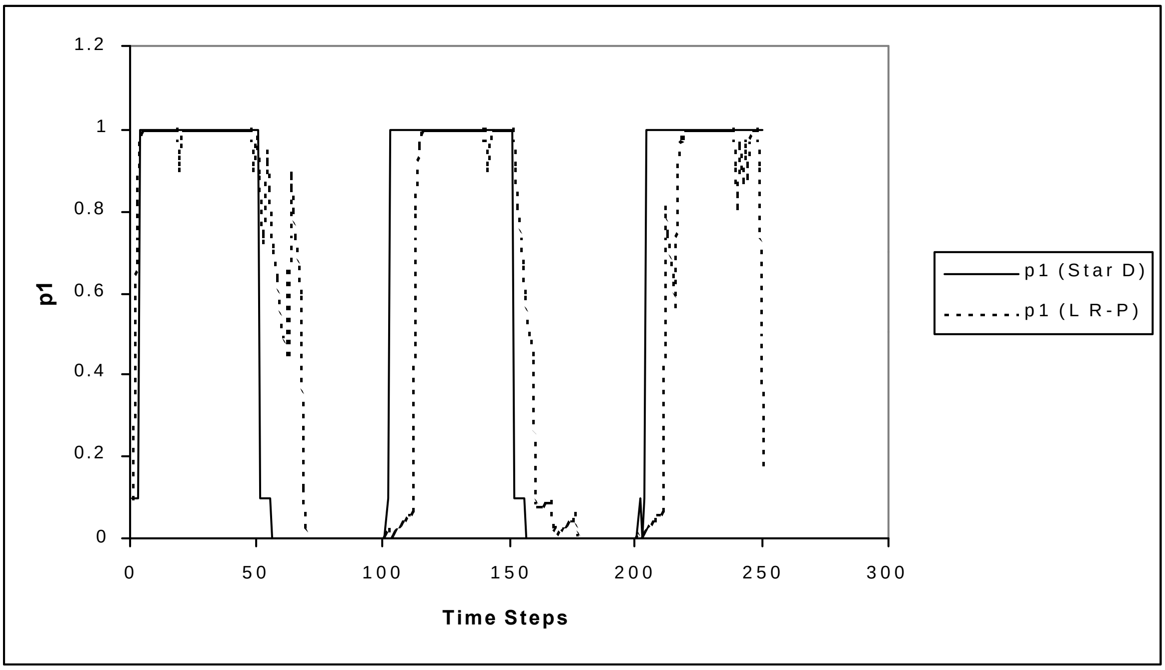

Figure 8b. Experiment nr.6. 250 time steps profiles of STAR$^{(D)}$ $(D = 2)$ p_1 probability (solid line) and $L_{R-\epsilon P}$ $(a = 0.90, b = 0.09)$ p_1 probability (dotted line).

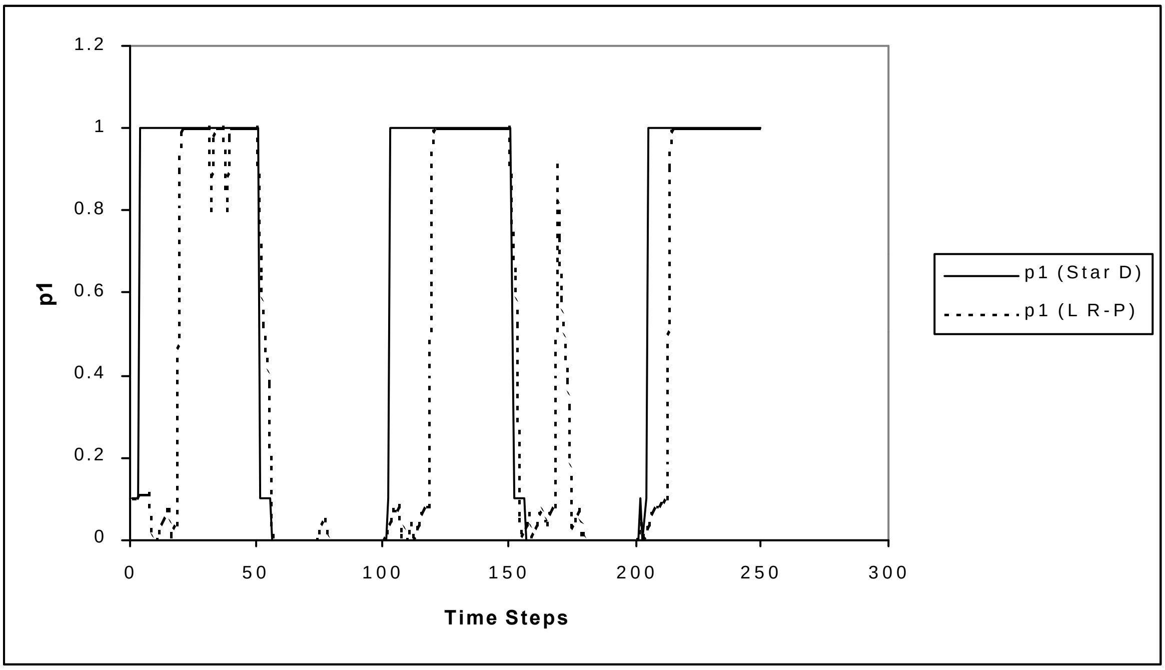

Figure 8c. Experiment nr.6. 250 time steps profiles of STAR$^{(D)}$ ($D = 2$) p_1 probability (solid line) and $L_{R-\epsilon P}$ ($a = 0.990, b = 0.198$) p_1 probability. (dotted line).

The figures presented above, support the conclusion that STAR responds faster than L_{R-P} and $L_{R-\epsilon P}$ to environnment switchings. As we have already seen, in general it also incurs smaller average cost. In addition to these advantages, it should be noted that STAR implementation is simpler, since it requires no floating point computations.

6 Conclusions

In this paper we compared the performance of traditional L_{R-P} 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 reward and/or penalty makes STAR$^{(1)}$ perform better than L_{R-P}. Finally, when depth D is increased, STAR$^{(D)}$ achieves ϵ-optimal behavior both for deterministic and probabilistic reward and/or penalty. Of course, this behavior is similar to that of the $L_{R-\epsilon P}$ automaton. It is well known that $L_{R-\epsilon P}$ may perform suboptimally in nonstationary environments and one suspects that the same may be true of STAR$^{(D)}$. To test this conjecture we perform a number of computer experiments, comparing the performance of L_{R-P}, $L_{R-\epsilon P}$ and STAR$^{(D)}$ in nonstationary environments. The conclusion is that in every case STAR outperforms L_{R-P}; in most cases it also outperforms $L_{R-\epsilon P}$, 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 very important 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. First, they generally give better performance; second, they converge faster; third they are simpler to implement. We believe that our conclusions may stimulate renewed research in the field of FSSA which will result in high-performance, simple-implementation learning algorithms. The STAR$^{(D)}$ architecture may be a first step in this direction.

A Mathematical Appendix

A.1

Our first goal is to derive equations (13),(21), (22), (30), (31),(39) and (40), which describe the matrix P of state transition probabilities for the STAR$^{(1)}$ automaton. This matrix actually depends on two parameters, δ and ϵ, both of which lie in the interval [0,1]. When $\delta = 0$, then an enviromental penalty response is followed by deterministic state transitions; when $\delta > 0$ the state transitions are probabilistic. A similar situation appears for ϵ and state transitions following a reward enviromental response. It is then obvious that the cases $\delta =$ and $\epsilon = 0$ are special cases, which are subsumed in the general case $\delta \in$ [0,1], $\epsilon \in$ [0,1]. We will use this fact to simplify the derivation of the previously mentioned equations. Namely, we will take general ϵ and δ and will prove that the elements of P are given by eqs. (39) and (40); then we will take δ and / or ϵ equal to zero to derive eqs. (13),(21), (22), (30) and (31) 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

$$P_{00} = Prob(\Phi(n+1) = 0|\Phi(n) = 0) =$$

$$\sum_{i=1}^{r} Prob(\alpha(n) = i|\Phi(n) = 0) \cdot Prob(\beta(n) = 1|\alpha(n) = i) \cdot Prob(\Phi(n+1) = 0|\beta(n) = 1) +$$

$$\sum_{i=1}^{r} Prob(\alpha(n) = i|\Phi(n) = 0) \cdot Prob(\beta(n) = 0|\alpha(n) = i) \cdot Prob(\Phi(n+1) = 0|\beta(n) = 0) =$$

$$\sum_{i=1}^{r} \frac{1}{r} \cdot c_i \cdot (1-\delta) + \sum_{i=1}^{r} \frac{1}{r} \cdot (1-c_i) \cdot \epsilon. \tag{60}$$

Similarly, for $i = 1, ..., r$

$$P_{0i} = Prob(\Phi(n+1) = 0|\Phi(n) = 0) =$$

$$Prob(\alpha(n) = i|\Phi(n) = 0) \cdot Prob(\beta(n) = 1|\alpha(n) = i) \cdot Prob(\Phi(n+1) = i|\beta(n) = 1) +$$

$$Prob(\alpha(n) = i|\Phi(n) = 0) \cdot Prob(\beta(n) = 0|\alpha(n) = i) \cdot Prob(\Phi(n+1) = i|\beta(n) = 0) =$$

$$\frac{1}{r} \cdot c_i \cdot \delta + \frac{1}{r} \cdot (1-c_i) \cdot (1-\epsilon). \tag{61}$$

Eqs. (60) and (61) are equivalent to eq. (39). We also have for $i = 1, ..., r$

$$P_{i0} = Prob(\Phi(n+1) = i|\Phi(n) = i) =$$

$$Prob(\alpha(n) = i|\Phi(n) = i) \cdot Prob(\beta(n) = 1|\alpha(n) = i) \cdot Prob(\Phi(n+1) = 0|\beta(n) = 1) +$$

$$Prob(\alpha(n) = i|\Phi(n) = i) \cdot Prob(\beta(n) = 0|\alpha(n) = i) \cdot Prob(\Phi(n+1) = 0|\beta(n) = 0) =$$

$$1 \cdot c_i \cdot (1-\delta) + 1 \cdot (1-c_i) \cdot \epsilon \tag{62}$$

and

$$P_{ii} = Prob(\Phi(n+1) = i|\Phi(n) = i) =$$

$$Prob(\alpha(n) = i|\Phi(n) = i) \cdot Prob(\beta(n) = 1|\alpha(n) = i) \cdot Prob(\Phi(n+1) = i|\beta(n) = 1) +$$

$$Prob(\alpha(n) = i|\Phi(n) = 0) \cdot Prob(\beta(n) = 0|\alpha(n) = i) \cdot Prob(\Phi(n+1) = i|\beta(n) = 0) =$$

$$1 \cdot c_i \cdot \delta + 1 \cdot (1-c_i) \cdot (1-\epsilon). \tag{63}$$

Eqs. (62) and (63) are equivalent to eq. (40). Now, letting $\delta = 0$, ϵ arbitrary, from eqs. (62) - (63) we obtain eq. (30)-(31); letting $\epsilon = 0$, δ arbitrary, from eqs. (62) - (63) we obtain eq. (21)-(22); letting $\delta = 0$, $\epsilon = 0$, from eqs. (62) - (63) we obtain eq. (13) and we are done.

A.2

Our next goal is to obtain convenient closed form expressions for the equilibrium state and action probabilities of STAR$^{(}$1). This can be done in case either δ or ϵ or both are zero, but not when both are nonzero. In other words, we will obtain eqs.(14), (15), (23), (24), (32) and (33).

We first consider the case $\delta = 0$ and $\epsilon = 0$, in other words with eqs.(14)-(15). As explained in Section 3, $\Phi(n)$, the state process, is irreducible an daperiodic, hence ergodic, and it possesses a limiting equilibrium (stationary) probability, call it 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 $i = 1, ..., r$

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

$$\pi_i = \pi_0 \cdot \frac{1}{r}\frac{1-c_i}{c_i}.$$

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

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

$$\pi_0 = \frac{r}{\sum_{j=1}^{r} \frac{1}{c_i}} \qquad \pi_i = \frac{1}{\sum_{j=1}^{r} \frac{1}{c_i}} \cdot \frac{1-c_i}{c_i} \qquad i = 1, ..., r.$$

which are exactly eq.(14). To obtain eq.(15), we observe that action i can only be taken when in state 0 or in state i. Hence, for $i = 1, ..., r$ we have

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

$$\frac{1 + \frac{1}{c_i} - 1}{\sum_{j=1}^{r} \frac{1}{c_j}}$$

which is exactly eq.(15).

To obtain eqs.(23) and (24) we only need to observe that, if in place of c_i we use $\hat{c}_i = c_i \cdot (1-\delta)$, eqs. (21) and (22), which define P, the matrix of state transition probabilities, take exactly the same form as eq. (13). Hence the equilibrium probabilities π and the action probabilities p are also of exactly the same form as eqs.(14), (15), except that we have $\hat{c}_i$ in place of c_i. This yields eqs. (23) and (24). Similarly, if we use in eqs. (30) and (31), $\bar{c}_i = c_i + \epsilon \cdot (1-c_i)$ we obtain exactly the same form as in eq. (13). This yields eqs. (32) and (33) and we are done.

A.3

We now turn to the study of the STAR$^{(D)}$ automaton. The computation of the state transition matrix $P^{(D)}$ follows exactly the same lines as for STAR$^{(1)}$, so it is not repeated. Let us now compute the equilibrium probabilities $\pi^{(D)}$. In other words we will obtain eqs.(56), (57), (58). In what follows we drop the superscript D for the sake of brevity.

Once again we start with the simplest case where $\delta = 0$, $\epsilon = 0$, in other words we will first obtain eq.(56). As explained in Section 4, $\Phi(n)$, the state process, is irreducible and aperiodic, hence ergodic, and it possesses a limiting equilibrium (stationary) probability, call it π. For the i-th branch of the star ($i = 1, ..., r$) we obtain terminal conditions

$$\pi_{(0,0)} \cdot \frac{1-c_i}{r} + \pi_{(i,2)} \cdot c_i = \pi_{(i,1)} \pi_{(i,D-1)} \cdot (1-c_i) + \pi_{(i,D)} \cdot c_i = \pi_{(i,D)} \tag{64}$$

and intermediate conditions

$$\pi_{(i,1)} \cdot (1-c_i) + \pi_{(i,3)} \cdot c_i = \pi_{(i,2)} \; ... \; \pi_{(i,D-2)} \cdot (1-c_i) + \pi_{(i,D)} \cdot c_i = \pi_{(i,D-1)}. \tag{65}$$

Combining the above equations we get

$$\pi_{(i,1)} = \frac{1}{r}\frac{1-c_i}{c_i}\pi_{(i,0)} \qquad \text{and} \qquad \pi_{(i,d)} = \frac{1-c_i}{c_i}\pi_{(i,d-1)} \qquad d=2,...,D. \tag{66}$$

From this it follows that

$$\pi_{(\,i,d)} = \big(\frac{1-c_i}{c_i}\big)^d \frac{\pi_{(i,0)}}{r} \qquad d=1,...,D. \tag{67}$$

Since $\pi_{(0,0)} + \pi_{(i,1)} + \ ... \ + \pi_{(r,D)} = 1$, we get

$$\pi_{(0,0)} \cdot \big(1 + \frac{1}{r}\sum_{d=1}^{D}(\frac{1-c_1}{c_1})^d + ...\frac{1}{r}\sum_{d=1}^{D}(\frac{1-c_r}{c_r})^d\big) = 1 \Rightarrow$$

$$\pi_{(0,0)} = \frac{r}{\sum_{j=1}^{r}\sum_{d=0}^{D}\big(\frac{1-c_j}{c_j}\big)^d} \tag{68}$$

$$\pi_{(i,d)} = \frac{r}{\sum_{j=1}^{r}\sum_{d=0}^{D}\big(\frac{1-c_j}{c_j}\big)^d} \cdot \big(\frac{1-c_i}{c_i}\big)^d \qquad i=1,...,r \qquad d=1,...,D. \tag{69}$$

These are the equilibrium state probabilities. To obtain the equilibrium action probabilities, sum the above equations over all states which produce action i, $i = 1,...,r$. These are states (0,0), $(i,1)$, ... , (i,D). We have

$$p_i = \pi_{(0,0)} \cdot \frac{1}{r} + \sum_{d=1}^{D}\pi_{(i,d)} = \frac{\sum_{d=0}^{D}\big(\frac{1-c_i}{c_j}\big)^d}{\cdot}\sum_{j=1}^{r}\sum_{d=0}^{D}(\frac{1-c_j}{c_j})^d \cdot \tag{70}$$

which is exactly eq.(56). For the cases $0 < \delta < 1$, $\epsilon = 0$ and $\delta = 0$, $0 < \epsilon < 1$ we use appropriate substitutions (just like for the case STAR$^{(1)}$) and obtain eqs.(57) and (58), respectively.

References

[1] R.R. Bush and F. Mosteller, *Stochastic Models and Learning*, J.Wiley & Sons, 1955.

[2] R.C. Atkinson and G.H. Bower, *An Introduction to Mathematical Learning Theory*, J. Wiley & Sons, 1965.

[3] K. Narendra and M.A.L. Thathatchar, *Learning Automata*, Prentice-Hall, 1989.

[4] M.F. Norman, *Markov Processes and Learning Models*, Academic Press, 1972.

[5] B.J. Oommen and E.R. Hansen "The asymptotic optimality of discretized linear reward-inaction learning automata", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-14, No. 3, pp. 542-545, May/June 1984.

[6] B.J. Oommen, "Absorbing and ergodic discretized two-action learning automata", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-16, No. 2, pp. 282-293, March/April 1986.

[7] B.J. Oommen and J.P.R. Christensen, "ϵ-optimal discretized linear reward-penalty learning automata", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-18, No. 3, pp. 451-458, May/June 1988.

[8] M.L. Tsetlin, "On the behavior of finite automata in random media", *Avtomatika i Telemekhanika*, Vol. 22, No. 10, pp. 1345-1354, Oct. 1961.

[9] V.I. Varshavskii and I.P. Vorontsova, "On the behavior of stochastic automata with a variable structure", *Avtomatika i Telemekhanika*, Vol. 24, No. 3, pp. 353-360, March 1963.