#3

Proceedings Inference OUIC 86

EDITORS

Danny R. Moates

Richard Butrick

OUIC 86

INTERDISCIPLINARY CONFERENCE ON INFERENCE

October 9-11, 1986

OHIO UNIVERSITY INTERDISCIPLINARY CONFERENCE ON INFERENCE

SPONSORED BY

THE COLLEGE OF ARTS AND SCIENCES

CONFERENCE BOARD

F. DONALD ECKELMANN, DEAN, COLLEGE OF ARTS AND SCIENCES

GENE BLOCKER	CHAIR, PHILOSOPHY
JAMES COADY	CHAIR, LINGUISTICS
THOMAS CREER	CHAIR, PSYCHOLOGY
KLAUS ELDRIDGE	CHAIR, COMPUTER SCIENCE
SHIH-LIANG WEN	CHAIR, MATHEMATICS

CONFERENCE DIRECTORS

RICHARD BUTRICK, CHAIR
ZINNY BOND, LINGUISTICS
DANNY MOATES, PSYCHOLOGY
MARMO SOEMARMO, LINGUISTICS
MARY ANNE SWARDSON, MATHEMATICS
LANGDON TAYLOR, COMPUTER SCIENCE
YIN-MIN WEI, COMPUTER SCIENCE
ARTHUR ZUCKER, PHILOSOPHY

PROGRAM CHAIRS

STEWART SHAPIRO
OHIO STATE, PHILOSOPHY
HAL R. ARKES
OHIO U., PSYCHOLOGY
MARIJKE AUGUSTEIJN
U. OF COLORADO, COMPUTER SCIENCE
WILLIAM FLEISSNER
U. OF PITTSBURGH, MATHEMATICS

RESULTS ON STRONGLY ABSOLUTELY EXPEDIENT LEARNING AUTOMATA

M. R. Meybodi

Computer Science Department
Ohio University
Athens, Ohio 45701

ABSTRACT

New results on Strongly Absoluely learning automata are reported.

I. Introduction

The concept of learning automata was introduced by Tsetlin [T2]. He investigated the learning behavior of finite automata under stationary random environment and showed that finite automata which have appropriate properties "behave well" if memory capacity tends to infinity. A great deal of work in the Soviet Union and elsewhere has followed the trend set by this source paper. The work of Tsetlin was later followed in the Soveit Union by various authors: Krylov[k1], Krinski [K2] and Ponomarev [p1]. However, it is practically impossible to build automata with unbound memory capacity. As an alternative, Varshavskii and Vorontsova introduced variable structure automata[v1]. In this setup, state transition probabilities or state probabilities of the finite automata is updated in accordance with reinforcement scheme or learning alogorithm.

The theory of learning automata is concerned with the analysis and synthesis of fixed or variable structure automata in a random environment.

Fixed Structure Automata: A fixed structure automata is a quintuple $\langle X,\phi,\alpha,F,G\rangle$

I. X is the input set of the automata. The input of the automata at the instant k denoted by x(k), is an element of set X. This set can be either a finite set or an infinite set, such as an interval on the real line ,then

$$X = \{x_1, x_2, \ldots, x_n\} \quad \text{or} \quad X = [a,b]$$

where a,b are real numbers.

II. ϕ is the set of states of the automata. The state of the automata at any instant k, denoted by $\phi(k)$, is an element of the finite set $\phi = \{\phi_1, \phi_2, \ldots, \phi_s\}$ $(2 \le s \le \infty)$

III. α is the output or reaction set of the automata. The action set of the automata at the instant k, denoted by $\alpha(k)$,is an element of finite set $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_H\}$ $(2 \le H \le \infty)$.

IV. The transition function $F: \phi \times X \to \phi$ determines the state at the instant (k+1) in terms of the state and output at time k,that is,

$$\phi(k+1) = F[\phi(k), X(k)].$$

Function F can be either deterministic or stochastic.

V. The output function $G: \phi \to \alpha$ determine the output of the automata at any instant in terms of the state at that instant,that is,

$$\alpha(k) = G(\phi(k)).$$

G could be a stochastic function but there is not loss of generality in assuming it to be deterministic [P2].

Remark 1: If the input set x of an automata contains only two elements 0 and 1 , it is called a P-model. If x takes more than two but a finite number of values, each with a positive probability, then we have a Q-model learning automata. If x takes on values in a continum, say for example $x \in [0,1]$, then we have an S-model learning automat.

Remark 2: The next state transition function F , may be specified by a transition matrix $[f^x_{ij}]$ for each $x \in X$. If these matrices contain only the elements 0 and 1 , the automata is a fixed structure deterministic automata. If the elements lie in the interval [0,1], the automata is a fixed structure stochastic automata. The input to the automata depends on the environment in which the automata is operating.

Remark 3: The output behavior of a learning automata for a given input sequence can be obtained if the transition probabilities are known. However, it may frequently be important to know the probability with which the automata is in a particular state at a given instant. These probabilities are known as state

probabilities and an alternative desciption of operation of the automata can be given in terms of these probabilities. At any time instant k, the automata being in state $\phi(k) = \phi_i(k)$ is governed by the probability distribution $\pi_i(k)$ such that

$$\sum_{i=1}^{s} \pi_i(k) = 1 \qquad \text{for all } k.$$

Let $\pi(k) = (\pi_1(k), \pi_2(k), \ldots, \pi_s(k))^T$, where T denotes the transpose and

$$\pi_i(k) = \text{Prob}[\phi(k) = \phi_i(k)].$$

One can also consider the action probability vector P(k) where the i^{th} component $p_i(k)$ is given by

$$p_i(k) = \text{Prob}[\alpha(k) = \alpha_i].$$

Variable Structure Automata: In fixed structure automata, the probabilities of state transitions are fixed. Variable structure automata update either the state transition probabilitties or equivalently the state probabilities on the basis of the output from the environment. The automata, in this case, is represented by the sixtuple $\langle x, \phi, \alpha, p, G, T \rangle$, where x is the input set, ϕ is the set of states, α is the action set, and p is the state probability vector governing the choice of the state at each stage (i.e. , at each stage k, $p(k) = (p_1(k), p_2(k), \ldots, p_s(k))^T$), and G is the output mapping. T is called an updating scheme or learning algorithm which is used to modify the state probability vector, that is, it generates p(k+1) from p(k).

Environment: The environment in which the automata operate is represented by the triple $\langle \alpha, x, D \rangle$. The environment has random response characteristics. It has input $\alpha(k) \in \{\alpha_1, \alpha_2, \ldots, \alpha_M\}$ and ouput belonging to the set x. In a P-model situtation, where $x(k) \in \{0,1\}$ the environment is characterized by the success probability vector $D = (d_1, d_2, \ldots, d_M)$ where

$$d_i = \text{Prob}[x(k) = 1 \mid \alpha(k) = \alpha_i].$$

If the d_i's don't depend on k, the environment is said to be stationary , otherwise it is nonstationary.

Let

$$d_1 \quad d_2 \quad d_3 \quad \ldots \quad d_M \qquad (1)$$

and $d_i \in (0,1)$ for all i = 1 to M. For them let $D = (d_1, d_2, \ldots, d_M)$.

So far we have considered the structure and properties of the environment and automata in isolation. Figure (1) represents a feedback connection of an automata and an environment. The actions of the automata from the inputs to the environment and responses or outputs of the environment in turn are the inputs to the automata. Starting from an initial state $\phi(0)$, the automata generates the corresponding action $\alpha(0)$. The response of the environment to this input is x(0), which in turn changes the state of the automata to $\phi(1)$. This sequence of operations is Then repeated to result in a sequence of states, actions and responses of the environment. In the case of Vriable Structure Learning Automata, the action probability vector p(k) (or the state transition matrix) gets updated at each state $\phi(k)$.

Norms of behavior: To judge the learning process objectively, it is necessary to set up quantitative norms of behavior. One quantity useful in judging the behavior of a learning automata is the average probability of success at time k given by

$$\eta(k) = E[x(k) \mid p(k)] = \text{Prob}[x(k) = 1 \mid p(k)]$$

$$= \sum_{i=1}^{M} \text{Prob}[x(k) = 1 \mid p(k), \alpha(k) = \alpha_i] \, \text{Prob}[\alpha(k) = \alpha_i]$$

$$= \sum_{i=1}^{M} d_i p_i(k).$$

If no prior information is available, there is no basis on which the different action α_i (i=1,2,.....,M) can be distinguished. In such a case, one would choose each action with equal probability (i.e.,by pure choice).In this case the value of average success is denoted by η_0 and is given

$$\eta_0 = \frac{1}{M} \sum_{i=1}^{M} d_i.$$

The use of the term learning automata can be justified if the average success is made greater than η_0 ,at least asymptotically. Such a behavior is called expedient and was introduced for the first time by Tsetlin [T2].

Definition 1: A learning automata is called expedient if

$$\lim_{k\to\infty} E[M(k)] > M_0 .$$

Definition 2: A learning automata is called optimal if

$$\lim_{k\to\infty} E[M(k)] = d_1 .$$

Definition 3: A learning automata is said to be ϵ-optimal if

$$\lim_{k\to\infty} | E[M(k)] - d_1 | < \epsilon$$

for any arbitrary $\epsilon > 0$ by a proper choice of the parameters of automata.

Definition 4: A learning automata is said to be absolutely expedient if

$$E[M(k+1) | p(k)] > M(k)$$

with probability one for all $p(k) \in S_m$ and for all D satisfying (1) with strict inequality holding good for all $p \in S_M^0$ where

$$S_M = \{ p \mid p=(p_1,p_2,\ldots,p_m)^M,\ 0 \le p_i \le 1,\ \sum_{i=1}^{M} p_i = 1 \}$$

and

$$S_M^0 = \{ p \mid p=(p_1,p_2,\ldots,p_M)^T,\ 0 < p_i < 1,\ \sum_{i=1}^{M} p_i = 1 \}.$$

Definition 5: A learning automata is said to be strongly absolutely expedient if

$$E[M(k+1) \mid p(k)] > M(k)$$

with probability one for all d satisfying (1) and for all $p \in S_M$ with strict inequality holding good for all $p \in (S_M - V_M)$ where

$$V_M = \{ e_i \mid i=1,2,\ldots,M \}$$

where $e_i = (0,0,0,0,\ldots,1,\ldots,0)$ be the i^{th} unit vector of dimention M.

Learning Algorithm: Variable structure automata updates either the transition probabilities or the action probabilities on the basis of the output from the environment. It is evident from the description of the learning automata that the crucial factor that affects the performance is the reinforcement scheme(learning algorithm) for updating the action probabilities.

A general scheme for updating the action probabilities can be represented as follows:

Let α_i be the action chosen at time k as a sample realization from the distribution p(k). Then p(k+1) is defined as follows:

$$p_i(k+1) = p_i(k) + \theta f_i^i[p(k)]$$

$$p_j(k+1) = p_j(k) - \theta f_j^i[p(k)] \quad j \neq i \text{ if success occurs}$$

and (1)

$$p_i(k+1) = p_i(k) - \theta f_i^i[p(k)]$$

$$p_j(k+1) = p_j(k) + \theta f_j^i[p(k)] \quad j \neq i \text{ if failure occurs}$$

where $f_j^i : S_M \to [0,1]$, $g_j^i : S_M \to [0,1]$, $i,j \in \{1,2,\ldots,M\}$ are nonnegative continuous functions and $0 < \theta < 1$ is called the step length parameter. The following consistancy condition

either $f_j^i[p] = 0$ for all $p \in S_M$

or $f_j^i[p] = p_j$ $j \neq i$ and

$$f_i^i[p] = \sum_{j \neq i} f_j^i[p] \qquad \text{(C1)}$$

and

either $g_j^i[p] = 0$ for all $p \in S_M$

$$\text{or} \qquad g_i^i[p] \neq p_i \quad \text{and} \qquad (C2)$$

$$g_i^i[p] = \sum_{j \neq i} g_j^i[p]$$

for all i,j = 1,2,...M, implies that $p(k+1) \in S_M$ if p(k) does. However, in order to make the algorithm nontrivial and interesting either $f_j^i \equiv 0$ or $g_j^i \equiv 0$ but not both. The continuity of f_j^i and g_j^i is one of mathematical convenience. The fact that both f and g are nonnegative maintains the reward and penalty nature of updating. If $f_j^i \neq 0$ but $g_j^i \equiv 0$ then (1) is called the reward-inaction algorithm, if $f_j^i \equiv 0$ but $g_j^i \neq 0$ then it is called the inaction-penalty algorithm, and if both f_j^i and g_j^i are nonzero then the algorithm is called the reward-penalty algorithm.

Let I= {1,2,3,....,M} and E = {success,failure}. The learning algorithm (1) define a mapping

$$T : S_M \times I \times E \to S_M \ ,$$

where

$$p(k+1) = T[p(k),i(k),e(k)] \ ,$$

where $i(k) \in I$ denotes the action chosen and $e(k) \in E$ the event occurring at time k. E is called the event space , p(k) and e(k) are known as the state and event sequence respectively. Let T^* be an extention of T defined as follows:

$$T^* : S_M \times (I \times E)^n \to S_M \quad \text{for all } n \geq 1 \ ,$$

where $(I \times E)^n$ refers to the n-fold cartesian product of I x E and

1) $T^* = T$ for n=1

2) $T^*[p,((i_0,e_0),(i_1,e_1),\ldots,(i_n,e_n))]$

$$= T[\ldots T[T[T[p,i_0,e_0],i_1,e_1],\ldots,i_n,e_n]],$$

where $i_j \in I$ and $e_j \in E$ for all j = 0,1,2,....,n and n-2. Define

$$K: S_M \times (I \times E) \to [0,1] \text{ as}$$

$$\text{Prob}[(i,e) \mid p] = K[p,(i,e)]$$

and

$$\text{Prob}[(i(k+1),e(k+1)) \mid p(0) = p \ ,(i(1),e(1)) \ ,l=0,1,\ldots k]$$

$$= K[p',(i(k+1),e(k+1))]$$

where

$$p' = T^*[p(0),\{(i(0),e(0)),(i(1),e(1)),\ldots,(i(k),e(k))\}].$$

Clearly

$$K[p,(i,e)] = p_i d_i \quad \text{if} \quad e = \text{success}$$

$$= p_i c_i \quad \text{if} \quad e = \text{failure}$$

That is K[p,.] is the event probability distribution.

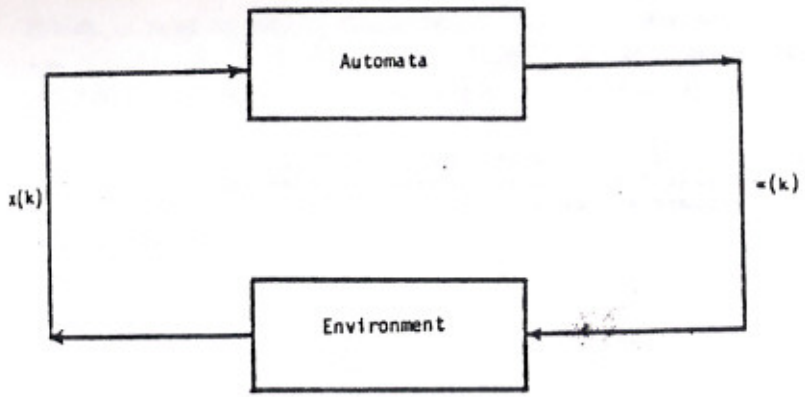
Definition 6: A state $p \in S_M$ is called an absorbing state of the algorithm T if

$$T[p,i,e] = p$$

for all $(i,e) \in I \times E$ with probability one. A learning algorithm is said to be absorbing if and only if there is at least one absorbibg state. If the absorbing states are the vertices of simplex S_M , the algorithm is called the absorbing barrier algorithm.

The basic idea behind a learning algorithm is a rather simple one. If the automata selects an action α_i at time k and success (x(k) = 1) results, the action probability $p_i(k)$ is increased and all the other components of vector p(k) are decreased. For failure (x(k) = 0) $p_i(k)$ is decreased and all the other components are increased.

Learning algorithm are generaly classified either on the basis of the behavior of the automata using the scheme, e.g., expedient, optimal, etc., or the nature of mapping T. That is , the nature of the function appearing in the algorithm,e.g., linear, nonlinear. If p(k+1) is a linear function of the component of p(k), the learning algorithm is said to be linear, otherwise it is nonlinear. It is also possible to classify the



Learning Automata

Figure 1

learning algorithm depending on whether it is an absorbing barrier algrithm or non-absorbing barrier algorithm(Ergodic algorithm). The non-absorbing barrier algorithms result in a Markov process with more than one absorbing barrier. Initial values of action probabilities do not affect the asymptotic behavior in the case of ergodic scheme, whereas in the case of absorbing barrier algorithm, the behavior of p(k) for large k is very much a function of initial values of the action probabilities. It can be shown [L3]that the distribution of p(k) generated by an ergodic algorithm after suitable normalization is normal; in other words, in the case of ergodic algorithm p(k) converges in distribution, but p(k) when generated by an absorbing barrier algorithm converges with probability one to a random vector whose range is a discrete(finite)set. It may be noted that a systematic study of ϵ-optimal Ergodic Algorithm is relatively new and for the first time discussed in the book by Lakshmivarahan [L3]. In [L3] a time varying algorithm where θ changes in time is discussed. Analysis of this latter class of algorithm depends on the stochastic approximation methods.

Various learning algorithm -- linear and non-linear, have been reported in the literature. The linear reward-penalty (L_{R-P}) scheme is one of the earliest scheme considered in mathematical psychology. The properties of this scheme have been studied in detail by a number of researchers in this field [B1,V1,M4,C1,V2]. It is known that an automata using linear reward-penalty scheme is expedient in all stationary random environment. The linear-inaction (L_{R-I}) scheme is another linear scheme which is derived by a modification of the L_{R-P} scheme. This scheme was considered first in mathematical psychology [B1] but was later independently conceived and intoduced into the engineering literature by Shapiro and Narenda [S1,S3]. The characteristics of the scheme is that it does not change the action probabilities whenever an unfavorable response resulted from the environment, following a favorable response, however, the probability of the action is increased as in L_{R-P} scheme. Because of this property, a learning automata using this scheme has been called a "benevolent automata" by Tsypkin and Poznyak [T4]. The L_{R-I} scheme was originaly reported to be optimal in all stationary random environment, but it is now known that it is only ϵ-optimal [V3, L6] L_{R-P} and L_{R-I} schemes can be considered as prototypes for the behavior encountered in many other complex schemes that have been investigated. Other possible combinations such as L_{R-R}, L_{P-P} and L_{I-P} have been studied in [V4]. It is shown that L_{I-P} scheme is always expedient only when an action is less penalized when it causes a nonpenalty response than the case when it causes a penalty response from the environment. L_{R-R} scheme is expedient if an action is more rewarded when it produces a penalty than if it produces a non-penalty.

The first nonlinear scheme for two state automata by Varshvskii and Vorontsova [V1] was in terms of transition probabilities. This can be used in the action probability version of the scheme. This scheme is ϵ-optimal in a restricted random environment satisfying either $d_1 < \frac{1}{2} < d_2$ or $d_2 < \frac{1}{2} < d_1$. Several nonlinear scheme which are ϵ-optimal in all random environment have been suggested by Viswanthan and Narenda [V2] , Sawaragi and Baba [S2], McMurtry and Fu [F2], Fu [F3] , and Lakshmivarahan and Thathacher [L2]. Instead of purely nonlinear and purely linear scheme, it is possible to combine them to get what is called hybrid scheme [V5].

In [A1], Aso and Kimura extended the class of the absolutely expedient learning algorithm. In the earlier works, the way of updating the probabilities used in selecting an action depended on reactions and dichotomy of actions: the selected and not selected; that is ,the functions that are used in updating are independent of the action chosen [M2,N1-N3,L2,L4,S2,V1]. In [A1] updating depends on which action is selected as well (Algorithm (1)). Interestingly, Aso and Kimura called this class of learning algorithm as "Stochastic Vector Automata" algorithm. The class of learning algorithms represented by scheme (1) is quite a general one and subsumes most of the known scheme avaiable. We give below some of the early schemes in term of general learning algorithm (1).

Scheme 1 (Linear Reward-Inaction Scheme): In algorithm (1) if we set $f_j^i[p] = cp_j$ and $g_j^i[p] = 0$ for all i, j, $i \neq j$, where $0<c<1$, we get the linear reward-inaction algorithm. This scheme is ϵ-optimal.

Scheme 2 (Beta Model): This scheme which was originally proposed by Luc [L1] in mathematical psychology as an alternative to Bush and Mosteller's L_{R-P} scheme (referred to as the α-model) can be obtained by setting

$$f_j^i[p] = \frac{(b-1)p_j(1-p_j)}{b(1-p_j) + p_j} ,$$

$$g_j^i[p] = \frac{(b-1)p_j}{(1-p_j) + bp_j} ,$$

where $b > 1$. This scheme is expedient in a restricted environment.

Scheme 3 (Vorontsova): This scheme can be obtained by setting

$$f_j^i[p] = a \neq (p_1,1-p_1)\, p_j^{\theta+1}(1-p_j)^{\theta} ,$$

$$g_j^i[p] = b \neq (p_1,1-p_1) p_j^{\theta+1}(1-p_j)^{\theta} ,$$

where

$$\neq (p_1,1-p_1) = \neq (1-p_1,p_1) .$$

This scheme applies only to a two-action automata. It is, however, the first nonlinear absolutely expdient scheme put forward, though at that time the concept of absolute expediency was unknown. This scheme is also ϵ-optimal.

Scheme 4 (Viswanathan and Narendra): This scheme can be obtained by setting

$$f_j^i[p] = p_j[a_1+a_2 p_j^{\theta+1}(1-p_j)^{\theta}] ,$$

$$g_j^i[p] = bp_j^{\theta+1}(1-p_j)^{\theta} ,$$

where the positive constants a_1, a_2 and b are to be chosen properly to satisfy consistency conditions (C1)-(C2). This scheme is absolutely expedient.

Scheme 5 (Lakshmivarahan and Thathacher): By setting $f_j^i[p] = \lambda(p)p_j$ and $g_j^i[p] = 0$ for all i and j, $i \neq j$, where $0<\lambda(p)<1$, and assuming that function $\lambda(p) = 0$ only if p is a unit vector we get the nonlinear reward-inaction reported in [L2]. This scheme is ϵ-optimal and absolutely expedient.

Aso and Kimura derived necessary and sufficient conditions for learning algorithm (1) to be absolutely expedient. Conditiins on functions g_j^i and f_j^i given by Aso and Kimura do not guarantee that V_M is the only set of absorbing barriers for the automata, and so they do not guarantee ϵ-optimality for the automata. At present there are a variety of algorithms which are ϵ-optimal. It is interesting to note that almost all of these algorithms have asymmetric behavior with respect to the occurrence of success and failure. It should be mentioned that the choice of functions f_j^i and g_j^i considered by Aso and Kimura do always induce identical behavior by the algorithm (1) under success and failure. Very recently Herkenrath et al. [H1] have derived necessary and sufficient conditions for the same class of learning algorithm considered by Aso and Kimura to be absorbing barrier algorithm. Meybodi and Lakshmivarahan [M3] derived necessary and sufficient conditions for a learning automata to be strongly absolutely expedient, and as a consequence they obtained a class of learning algorithm which has identical behavior under the occurrence of success and failure. The choice of functions given in [M3] induce identical behavior of algorithm 1 under the occurrence of success and failure. In this report more results on strongly absoluely expedient learning automat are reported. A number of computer simulation results are also given.

2. Convergence with Probability One

In this section conditions on algorithm 1 are discussed such that the Markov process $\{p(k)\}_{k\geq 0}$ converges with probability one. At first the consistency conditions (1.2) are rewritten in a form more suitable for the analysis. Let

$$f_i^i[p] = \alpha[i,p](1-p_i(k)) ,$$

$$f_j^i[p] = \alpha[i,j,p]p_j(k), \qquad (C.1)$$

$$\sum_{j\neq i} \alpha[i,j,p]p_j(k) = \alpha[i,p](1-p_i(k)) ,$$

and

$$g_i^i[p] = \lambda[i,p]p_i(k) \ ,$$

$$g_j^i[p] = \delta[i,j,p](1-p_j(k)) \ , \qquad (C.2)$$

$$\lambda[i,p]p_i(k) = \sum_{j \neq i} \delta[i,j,p](1-p_j(k)) \ ,$$

where $\alpha, \lambda : I \times S_m \to [0,1]$ and $\beta, \delta : I \times I \times S_m \to [0,1]$. In view of conditions (C.1) and (C.2), the calss of updating scheme which is considered in this chapter is defined as follow:

Let α_i be the action chosen at time k as a sample realization from the distribution p(k). Then p(k+1) is defined as follows:

$$p_i(k+1) = p_i(k) + \theta\alpha[i,p](1-p_i(k)) \ ,$$

$$p_j(k+1) = p_j(k) - \theta\beta[i,j,p]p_j(k) \ , \quad j \neq i$$

if the action chosen resulted in success, and (2)

$$p_i(k+1) = p_i(k) - \theta\lambda[i,p]p_i(k) \ ,$$

$$p_j(k+1) = p_j(k) + \theta\delta[i,j,p](1-p_j(k)) \ , \quad j \neq i \ ,$$

if the action chosen resulted in failure, where $0 < \theta < 1$ is step length parameter.

The basic rule that governs the choice of fuctions in the above algorithm (algorithm 2) is that if a term is subtracted from p_j , then it is made proportional to p_j , and if a term is added to p_j, then it is made proportinal to $1-p_j$ irrespective of which action is chosen and whether the action results in success or failure.

The following theorem gives a set of necesary and sufficient conditions for the algorithm (2) to be an absorbing barrier learning algorithm.

Theorem 1: Necessary and sufficient conditions for the learning algorithm (2) to have V_M as the only set of absorbing states are:

(A.1) For all $p \in S_m - V_M$ there exists $1 \leq s \leq M$ such that

$$[\alpha[s,p] + \lambda[s,p]]p_s > 0$$

(A.2) For all $1 \leq s \leq M$, $\lambda[s,e_s] = 0$.

Proof: Refer to [M9].

Remark 1: From theorem 1 it can be observed that if $p \in V_M$ then $p = e_s$ is an absorbing state if and only if $f_j^s[e_s] = g_j^s[e_s] = 0$ for all $1 \leq j \leq M$; if $p \in S_M - V_M$, then p is absorbing if and only if $f_j^s[p]p_s = g_j^s[p]p_s = 0$ for all $1 \leq s, j \leq M$.

The typical choice of functions in algorithm (2) that satisfies the conditions (C.1) - (C.2) and (A.1) and (A.2) are given in the following examples.

Example 1:

$$\beta[i,s,p] = a_1(1-p_i)(1-p_s)$$

$$\alpha[i,p] = a_1 \sum_{s \neq i} p_s(1-p_s)$$

$$\delta[i,s,p] = a_2 p_i p_s^2 (1-p_i)$$

$$\lambda[i,p] = a_2)(1-p_i) \sum_{s \neq i} p_s^2(1-p_s)$$

for all $i,s = 1,2,\ldots,M$ where $0 < a_1, a_2 < 1$.

Example 2:

$$\beta[i,s,p] = a_1 (1-p_i)^n(1-p_s)$$

$$\alpha[i,p] = a_1(1-p_i)^{n-1}\sum_{s\neq i} p_s(1-p_s)$$

$$\delta[i,s,p] = a_2 p_i^{m} p_s^{2}(1-p_i)$$

$$\lambda[i,p] = a_2 p_i^{m-1}(1-p_i)\sum_{s\neq i} p_s^{2}(1-p_s)$$

for all $i,s = 1,2,\ldots,M$ where $0<a_1,a_2<1$, $m,n > 0$

Example 3:

$$\beta[i,s,p] = a_1(1-p_i)^{n}(1-p_s)^{n}$$

$$\alpha[i,p] = a_1(1-p_i)^{n-1}\sum_{s\neq i}(1-p_s)^{n}p_s$$

$$\delta[i,s,p] = a_2 p_i^{m-1} p_s^{m}(1-p_i)$$

$$\lambda[i,p] = a_2 p_i^{m-2}(1-p_s)\sum_{s\neq i} p_s^{m}(1-p_s)$$

for all $i,s = 1,2,\ldots,M$ where $0< a_1,a_2<1$, $n,m>0$

Example 4 :

$\beta[i,s,p] \equiv 0$ for all i,s

$\alpha[i,s] \equiv 0$ for all i

$$\delta[i,s,p] = a_s^{i}(1-p_i)p_s$$

$$\lambda[i,p] = \begin{cases} \dfrac{1-p_i}{p_i}\displaystyle\sum_{s\neq i} a_s^{i} p_s(1-p_s) & \text{if } p_i > 0 \\ 0 & \text{if } p_i = 0 \end{cases}$$

for all $i,s = 1,2,\ldots,M$, where $0 < a_s^{i} < 1$ for all i,s and $a_s^{i}=a_i^{s}$.

Example 5:

$$\beta[i,s,p] = a_1(1-p_s)$$

$$\alpha[i,p] = \begin{cases} a_1\dfrac{\sum(1-p_s)p_s}{1-p_i} & \text{if } p_i < 1 \\ 0 & \text{if } p_i = 1 \end{cases}$$

$\delta[i,s,p] \equiv 0$

$\lambda[i,p] \equiv 0$

for all $i,s = 1,2,\ldots,M$ where $0<a_1<1$.

Example 6:

$\beta[i,s,p] = 0$ for all i,s

$\alpha[i,p] \equiv 0$ for all i

$$\delta[i,s,p] = a_s^{i}(1-p_i)p_s B_s^{i}$$

$$\lambda[i,p] = \begin{cases} \dfrac{\displaystyle\sum_{s\neq i} a_s^{i}(1-p_i)p_s B_s^{i}}{p_i} & \text{if } p_i > 0 \\ 0 & \text{if } p_i = 0 \end{cases}$$

for all $i,s = 1,2\ldots,M$ where $b: I\times I\times S_M \to [0,1]$ and

$a_s^i = b_i^s = 0$ for all i,s ,and $0 < a_s^i < 1$ and $a_s^i = a_i^s$ for all s and i.

Example 7:

$$\lambda[i,j,p] = 0$$

$$\alpha[i,p] = 0$$

$$\delta[i,j,p] = a_j^i(1-p_j)p_j[b_j^i p_i(1-p_j) + b_i^j p_j(1-p_i)]$$

$$\lambda[i,p] = \begin{cases} \dfrac{(1-p_i)\sum_{j\neq i} a_j^i p_j[b_j^i(1-p_j)p_i + b_i^j p_j(1-p_i)]}{p_i} & \text{if } p_i > 1 \\ 0 & \text{if } p_j = 0 \end{cases}$$

for all i,j = 1,2,...,M where $0 < a_j^i , b_j^i < 1$ for all i,j and $a_j^i = a_i^j$ for all i,j, $i \neq j$.

Theorem: Necessary and sufficient conditions for the learning algorithm (2) to be strongly abolutely expedient are

$$\sum_{j\neq i} p_j \lambda[i,j,p] = \sum_{j\neq i} p_j \lambda[j,i,p] \qquad (S.1)$$

and

$$\sum_{j\neq i} p_i(1-p_j)\delta[i,j,p] = \sum_{j\neq i} p_j(1-p_i)\delta[j,i,p] \qquad (S.2)$$

for all i,j = 1,2,...,M.

Among the 7 examples given above , examples 1,3,4,6 and 7 not only saisfy conditions (C.1)-(C.2) and (A.1)-(A.2) but they satisfy conditions (S.1)-(S.2) and hence they are strongly absolutely expedient. Examples 2 and 5 are examples of of non absolutely expedient but absorbing learning algorithms.

Remark 5: Define $\Delta x(k) = x(k+1) - x(k)$, and let

$$\Delta u(k) = E[\Delta u(k) | p(k)] = \sum_{i=1}^{M} E[\Delta p_i(k) | p(k)]d$$

It can be seen by direct computation that

$$E[\Delta p_i(k) | p(k)=p] = p_i(1-p_i)d_i\alpha[i,p] - p_i^2 c_i \lambda[i,p] - \sum_{j\neq i} p_j p_i d_j \lambda[j,i,p] + \sum_{j\neq i} p_j(1-p_i)c_j\delta[j,i,p]. \qquad (3)$$

Remark 6: By setting $\delta[i,j,p] = 0$ and $\lambda[i,j,p] = c$ $(0<c<1)$ for all i,j, where c is constant, we get linear reward-inaction scheme. It is easy to verify that this is the only linear sch that satisfies conditions (C.1)-(C.2),(A.1)-A.2) and (S.1)-(S In fact this scheme is the only one among all the linear sch which is strongly absolutely expedient. Linear reward-inactio scheme can be obtained by setting n=0 and a_2=0 in example 3.

Remark 7: By setting $\lambda[i,j,p] = \rho(p)$ and $\delta[i,j,p] = 0$ for all where $0 < \rho(p) < 1$, and assuming that the function $\rho(p) = 0$ onl p is a unit vector, we get the reward-inaction scheme reporte [L2]. This scheme satisfies the conditions (C.1)-(C.2) and conditions of theorem 1 and 2, and therefor is srongly absolu expedient.

Remark 8: If $\lambda[i,j,p] = (1-p_j)(1-p_j)\rho_{ij}(p)$ and $\delta[i,j,p] = 0$ for all i,j = 1,2,...,M, where $\rho_{ij}(p) = \rho_{ji}(p)$ and $\rho_{ij}(p) : S$ [0,1] and if for all $p \in S_M - V_M$ there exist i and j such $\rho_{ij}(p$ then algorithm 2 is strongly absolutely expedient.

Proof of this remark is immediate from theorem 1 and 2. Clearly (C.1)-(C.2) and (S.1)-(S.2) are satisfied and (A.1) i satisfied if for all $p \in S_M - V_M$ there exist i and j such that

$\alpha_{ij}(p) \quad 0.$

Theorem 2 established the equivalence of strong absolute expediency and conditions (S.1)-(S.2). The following theorem in turn proves the equivalence of conditions (S.1)-(S.2) and the sign definitness of Δp_1 and Δp_M. Δp_i is defined to be

$$\Delta p_i(k) = E[\delta p_i(k) \mid p(k)]$$

and α_1 and α_M are the actions associated with the largest and smallest success probability, respectively.

Theorem 3: When a learning automata uses learning algorithm 2 and operates in any stationary random environment, conditions (S.1)-(S.2) are necessary and sufficient conditions for $\Delta p_1(k) > 0$ and $\Delta p_M(k) < 0$ for all $p(k) \in S_M - V_M$.

Sufficiency: Using (S.1)-(S.2) ,(3) canbe rewritten as

$$\Delta p_i(k) = \sum_j (d_i - d_j) H_{ij}$$

where

$$H_{ij} = p_j(k)\alpha[i,j,p]p_i(k)\delta[j,i,p](1-p_i(k))$$

The sufficiency is immediate, since for any j, $d_i - d_j \geq 0$ if i = 1 and for any j, $d_i - d_j \leq 0$ if i= M and $H_{ij} \geq 0$ for all i,j = 1, 2,...,M.

Necessity: (3) can be rewritten as

$$\Delta p_i(k) = - r_i + (r_i + s_i)d_i + \sum_{j \neq i} (d_i - d_j) H_{ij}$$

where

$$r_i = p_i(k)\sum_{j \neq i} \alpha[i,j,p](1-p_j(k)) - \sum_{j \neq i} p_j\delta[j,i,p](1-p_i(k))$$

and

$$s_i = p_i(k) \sum_{j \neq i} [\delta[i,j,p](1-p_j(k)) + \alpha[i,j,p]p_j(k)]$$

$$- \sum_{j \neq i} p_j(k)[\delta[j,i,p](1-p_i(k)) + \alpha[j,i,p]p_i(k)]$$

Let us fix the subscript i. Consider two environments $D^{(1)} = (d_1^{(1)}, d_2^{(1)}, \ldots, d_M^{(1)})$ and $D^{(2)} = (d_1^{(2)}, d_2^{(2)}, \ldots, d_M^{(2)})$ such that $d_i^{(1)} = d_i^{(2)} = x$, $d_j^{(1)} = y$ $(j \neq i)$, and $d_j^{(2)} = z$ $(j \neq i)$ where $1 \geq y \geq x \geq z \geq 0$. Let the corresponding Δp_i's be denoted by $\Delta p_i^{(1)}(k)$ and $\Delta p_i^{(2)}(k)$. Then

$$\Delta p_i^{(1)}(k) = - r_i + (r_i + s_i)x + \sum_{j \neq i} (x-y) H_{ij} \qquad (4)$$

$$\Delta p_i^{(2)}(k) = - r_i + (r_i + s_i)x + \sum_{j \neq i} (x-z) H_{ij}$$

From (4) it follows that $\Delta p_i(k) > 0$ and $\Delta p_M(k) < 0$ are equivalent to

$$-(y-z) \sum_{j \neq i} H_{ij} < r_i - (r_i+s_i) < (x-z)\sum_{j \neq i} H_{ij} .$$

Since the inequality holds for any x,y,z, it should hold if y=x and z=x. Hence we have $r_i - (r_i + s_i)x = 0$. In particular, this implication should hold for x=0 and x=1. Then we have $r_i = 0$ and $s_i = 0$ as x=0 and x=1 respectively. Since the above holds for any i, the proof is immediate.

Remark 8: The sign definiteness of $\Delta p_i(k)$ ($\Delta p_M(k)$) implies that $E[p_i(k)]$ ($E[p_M(k)]$) is monotonically increasing (decreasing) with respect to k. Theorem 2 thus states that by using absolutely expedient schemes $p_i(k)$ and $p_M(k)$ we move in the right direction at least in an expected-value sense.

Corollary 1: The learning algorithm (2) is strongly absolutely expedient if and only if

$$\Delta p_1(k) > 0 \quad \text{and} \quad \Delta p_M(k) < 0$$

for all $p \in S_M - V_M$.

Proof: The proof is obvious from Theorem 2 and 3.

Remark 10: Making $\Delta p_1(k) > 0$ and $\Delta p_M(k) < 0$ implies that $p_1(k)$ is a sub-martingle and $p_M(k)$ is super-martingle.

In view of martingle theorem[D1] we can conclude that

$$\lim_{k\to\infty} p_1(k) = p_1^*$$

and

$$\lim_{k\to\infty} p_M(k) = p_M^*$$

exist with probability one. Since $\Delta p_1(k) = 0$ when $p_1 = 0$ or 1, p_1^* and p_M^* approaches 0 or 1, each with positive probability.

Corollary 2: The Markov process $\{p(k)\}$ $k \geq 0$, as generated by the algorithm (2) under conditions (A.1)-(A.2), and (S.1)-(S.2) converges to V_M with probability one.

Proof: Let $F[x(1),x(2),\ldots,x(k)]$ be the σ-algebra generated by a random sequence $x(1),x(2),\ldots,x(k)$. As $n(k)$ is a linear function of $p(k)$, it easily follows that

$$F[n(1),n(2),\ldots,n(k)] \subset F[p(1),p(2),\ldots,p(k)]. \quad (5)$$

Further

$$F[p(k)] \subset F[p(1),p(2),\ldots,p(k)]. \quad (5a)$$

From theorem (2) we know that

$$E[\delta n(k) \mid p(k)] = \Delta n(k) = \frac{1}{4} \sum_{i=1}^{M} \sum_{j\neq i} p_i p_j (d_i - d_j)^2 \{\alpha[i,j,p] + \alpha[j,i,p]\} + \frac{1}{4} \sum_{i=1}^{M} \sum_{j\neq i} p_i (1-p_i)^2 \{\beta[i,j,p] + \beta[j,i,p]\} \geq 0 . \quad (6)$$

Using (5a) we get

$$E[n(k+1) \mid F[p(1),p(2),\ldots,p(k)]] \geq n(k) .$$

Now taking conditional expectation on both sides of (6) with respect to $F[n(1),n(2),\ldots,n(k)]$ from the law of iterated conditional expectations and from (5) we get

$$E[n(k+1) \mid f[n(1),n(2),\ldots,n(k)]] \geq n(k) .$$

This in turn implies that $n(k)$ is a non-negative bounded sub-martingle and by martingle theory $n(k)$ converges to a rand variable n^* with probability one, and hence $\lim_{k\to\infty} p(k) = p^*$ exists with probability one. As $\delta n(k) = 0$ only on V_M, it foll that $p^* \in V_M$ with probability one. This implies that the set of all unit vectors are the only absorbing barriers for the proce $\{p(k)\}$.

The above theorem stated that $\{p(k)\}$ corresponding to algorithm (2) under conditions (A.1)-(A.2)and (S.1)-(S.2) is a process with more than on absorbing barriers, that is

$$\lim_{k\to\infty} p(k) \in \{e_i : i = 1,2,\ldots,M\} \text{ with probability one.}$$

out of all these M unit vectors only one corresponds to the optimal or desired action. In fact, the Markov process $\{p(k)\}$ converges with a positive probability to each one of these uni vectors. Therefore , strong absolute expediency implies that t algorithm (2) converges to the optimal action with a positive probability less than one.

The following theorem states that strong absolute expedi is a sufficient condition for ϵ-optimality.

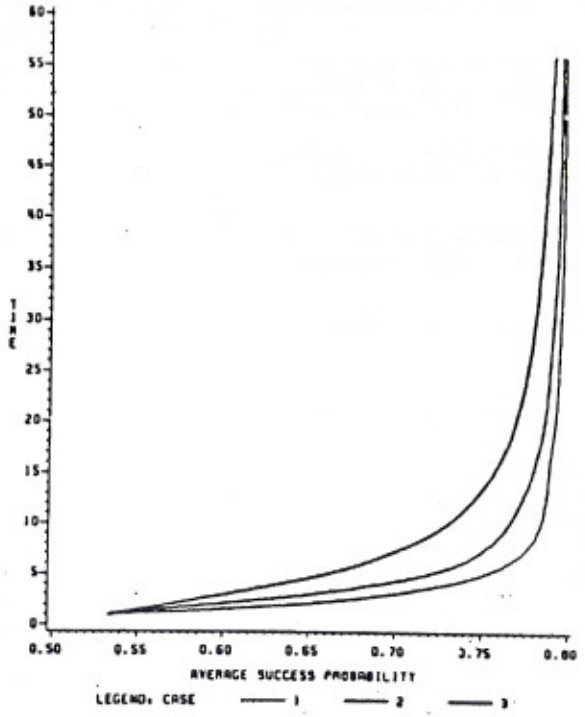
Teorem 4: For every $\epsilon > 0$ and $p(0) = p \in S_M^0$ there exists $\theta^* > 0$ such that for all $0 < \theta < \theta^*$, the learning algorithm under the conditions (C.1)-(C.2), (A.1)-(A.2), and (S.1)-(S.2) is such that

$$\lim_{k\to\infty} | E[m(k)] - d_1 | = 0 .$$

Proof: Refer to [M3].

SIMULATIONS: The algorithm (2) was simulated for the choice of the functions given in example 1. The average value of $m(k)$ averaged over 20 sample runs for various values of θ are given in figure 2.

EXAMPLE 1



Case #1 $\theta = 0.2$

Case #2 $\theta = 0.1$

Case #3 $\theta = 0.05$

Figure 2

References

[A1] Aso,H. and Kimura, M., "Absolute Expediency of Learning Automata." Information Science, Vol. 17,pp. 91-112,1979.

[B1] Bush, R.R. and Mosteller,F., Stochastic Models for Learning. Wiley, 1958.

[B2] Baba, R.R. and Sawaragi, Y., " On Learning Behavior of Stochastic Automata under Non-Stationary Environment." IEEE Tansactions on System ,Man and Cybernetics, Vol. 5,PP.273-275, 1975.

[C1] Chandrasekran,B. and Shen, D.W.C., " On Expediency and Convergence in Variable Structure Automata." IEEE Transactions on System Science and Cybernetics, Vol. 4, pp. 52-60, 1968.

[D1] Doob, J. L. ,Stochastic Processes. Wiley, 1951.

[F1] Fu,K. S. , " Learning Control System. Review and outlook." IEEE Tansactions on Automatic Control, Vol. 15, pp. 210-221, April 1970.

[F2] Fu, K. S. , " Learning Control System and Intelligent Control System: An intersection of Artificial Intelligence and Automatic Control." IEEE Transactions on Automatic Control, Vol. 16, pp. 70-72,1971.

[F3] Fu, K. S., " Stochastic Automata Models for Learning Systems." In Computers and Information SCience II (Ed) by J. T. Tou, Acamedic, 1967.

[H1] Herkenrath, U., Kalin, D. and Lakshmivarahan,S., " On a General Class of Absorbing Barrier Learning Algorithm. " Information Science, Vol. 23,1981.

[k1] Krylov, V. Yu., " On One Stochastic Automata which is Asymptotically Optimal in a Random Media." Automata and Remote Control, Vol. 24,pp. 1114-1116, 1963.

[K2] Krinskii, V. I. , " An Asymptotically Optimal Automata with Exponential Convergence." Bio Physics, Vol. 9, pp. 484-487, 1984.

[L1] Luce ,R.D., Individual Choice Behavior. Wiley, 1959.

[L2] Lakshmivarahan, S. and Thathachar, M.A.L., " Absolutely Expedient Learning Algorithm for Stochastic Automata." IEEE transactions on Systems, Man and Cybernetics, Vol. 3,pp. 281-286,1973.

[L3] Lakshmivarahan, S. , Learning Algorithms: Theory and Apllications. Springer Verlag, New York, 1981.

[L4] Lakshmivarahan, S. and Thathachar, M.A.L., " Bounds on the Probability of Convergence of Learning Automata." IEEE Transactions on Systems , Man and Cybernetics, Vol. 6,pp.756-753,1976.

[L5] Lakshmivarahan, S. and Thathacher, A.L., " Absolutely Epedient Learning Algorithms for Stochastic Automata." IEEE Transactions on Systems, Man and Cybernetics, Vol. 3. pp. 281-286, 1973.

[L6] Lakshmivarahan, S. " Learning Algorithms for Stochastic Automata." Ph.D. Dissertation, Dept. Elec. Engr. , India Institute of Science, Bangalore, India, Jan. 1973.

[M1] Mendel,J.M. and Fu,K.S., Adaptive Learning and Pattern Recognition Systems. Academic Press, 1970.

[M2] McMurtry, G. J. and Fu,K. S. , " A Variable Structure Automata Used as a Multi-model Search Technique. " IEEE Transactions on Automatic Control, Vol. 11,pp. 379-387,1966.

[M3] Meybodi, M. R. and Lakshmivarahan S. , " ε- Optimality of a General Class of Learning Algorithm." Information Science, Vol. 28, pp. 1-20,1982.

[N1] Norman, M. F., " On Linear Models with Two Absorbing Barriers." Journal of Mathematic Psychology. Vol. 5,pp. 225-241,1968.

[N2] Norman, M. F. , Markov Processes and Learning Models. Academic Press,1972.

[N3] Narendra , K. S. and lakshmivarahan, S., " Learning Automata-A Survey." IEEE Trans. System, Man and Cybernetics, Vol.4 pp.323-334,1974.

[N4] Narendra, K. S. and Lakshmivarahan, S. ,"Learning Automata, A Critique." Journal of Cybernetics and Information Sciences-Special Issue on Learning Automata. Vol. 1, pp. 53-66, 1978.

[N5] Narenda, K. S., Wright, E. and Mason, L. G., " Application of Learning Automata to Telephone Traffic Routing. " IEEE Trans. on Systems, Man and Cybernetics, oo. 785-792, 1977.

[N6] Narendra, K. S. and THathacher, M. A. L. , " On the Behavior of a Learning Automata in a Changing Environment with Routing Applications." IEEE Trans. on System, Man and Cybernetics, Vol. 10, pp. 262-269,1980.

[N7] Narendra, K. S. and Wiswanathan, R. , " A two - Level System of Stochastic Automata for Periodic Random Environment." IEEE Trans. on System , Man and Cybernetics, Vol. SMC-2, pp. 285-289, 1972.

[N8] Norman, M. F. , "Some Convergence Theorems for Stochastic Learning Models with Distance Diminishing Operators." Journal of Mathematical Psychology, Vol. 5, pp. 61-101,1968.

[N9] Narendra, K. S. and Thathachar, M.A.L., Learning Automata, forthcoming book.

[P1] Ponomarov, V. A., " A Construction of an Automata which is Asymptotically Optimal in a Stationary Random Media. " Bio Physics, Vol. 9, pp. 104-110,1964.

[S1] Sklansky, J., " Learning System for Automatic Control." IEEE Trans. on Automatic Control, Vol. 11, pp. 6-19, 1966.

[S2] Sawargi, Y. and Baba, N., "Two ε-optimal Non-linear Reinfocement Scheme for Sthcahastic Automata." IEEE Trans. on System, Man and Cybernetics, Vol.4, pp. 126-131, 1974.

[S3] Shapiro, I.J., " The Use of Stochastic Automata in Adaptive Control. " Ph.D. Dissertation, Dept. Engr. and Applied Sci.,Yale Univ., New haven, Conn., 1969.

[T1] Tsypkin, Ya Z. , Adaptation and Learning in Automatic System, Academic Press, 1971.

[T2] Tsetlin, M. L.,Automation Theory and Modeling of Biological System. Academic Press, 1973.

[T3] Tsypkin, Ya Z., Foundations of the Theory of Learning Systems, Academic Press, 1973.

[T4] Tsypkin, Ya Z. and Poznyak, A. S., " Finite Learning Automata." Engr. Cyber., Vol. 10, pp. 478-490, May-june 1972.

[V1] Varshavskii, V. I. and Vorontsova, I. P., " On the Behavior of Stochastic Automata with Variable Structure." Automation and Rmote Control, Vol. 24, pp. 327-333,1963.

[V2] Viswanathan, R. and Naredra, K. S., " Expedient and Optimal Variable Structure Stochastic Automata." Dunham Lab. TR-37, 1970. Yale Univ. New haven, Conn.

[V3] Viswanathan, R. and Narendra, K. S., " Anote on Linear Reinforcement Scheme for Variable Structure Stochastic Automata." IEEE Trans. on Systems, Man and Cybernetics, Vol. 4, pp. 131-135,1974.

[V4] Viswanathan, R., " Learning Automata: Model and Application." Ph.D. Dissertation, Dept. Elec. Engr. and Applied Sci., Yale Univ., New Haven, Conn., 1972.