ABSORBING AND ERGODIC DISCRETIZED TWO ACTION LEARNING AUTOMATA

B. John Oommen

SCS-TR-74

May 1985

School of Computer Science

Carleton University

Ottawa, Ontario

K1S 5B6

Canada

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada.

ABSORBING AND ERGODIC DISCRETIZED TWO ACTION LEARNING AUTOMATA[+]

B. John Oommen[*]

ABSTRACT

A learning automata is a machine that interacts with a random environment and which simultaneously learns the optimal action which the environment offers to it. In this paper we consider learning automata which have a variable structure. Such automata are completely defined by a set of probability updating rules [4,9,20]. All the Variable Structure Stochastic Automata (VSSA) discussed in the literature, update the probabilities in such a way that an action probability can take any real value in the interval [0,1]. As opposed to these, in this paper we shall discretize the probability space so as to permit the action probability to assume one of a finite number of distinct values in [0,1]. The discretized automaton is termed linear or nonlinear depending on whether or not the sub-intervals of [0,1] are of equal length. We shall prove that:

(1) Discretized Two-Action Linear Reward-Inaction Automata are absorbing and ε-optimal in all environments.

(2) Discretized Two-Action Linear Inaction-Penalty Automata are ergodic and expedient in all environments.

---

* Partially supported by the Natural Sciences and Engineering Research Council of Canada.

+ School of Computer Science, Carleton University, Ottawa, Canada, K1S 5B6.

A preliminary version of this paper will be presented at the 1985 International Conference on Systems, Man and Cybernetics, Tucson, Arizona.

(3) Discretized Two-Action Linear Inaction-Penalty Learning Automata with artificially created absorbing barriers are ε-optimal in all random environments.

(4) There exist nonlinear discretized Reward-Inaction Automata which are ε-optimal in all random environments. Further, the maximum advantage gained by rendering any finite-state discretized automaton nonlinear has also been derived.

Apart from the above theoretical results various simulation results will be presented which seem to indicate that the ergodic two-action linear Reward Penalty automation is ε-optimal in all environments. The latter demonstrates powerful learning capabilities when interacting with non-stationary environments.

I. INTRODUCTION

Learning automata have been extensively studied by researchers in the area of adaptive learning. The intention is to design a learning machine which interacts with an environment and which dynamically learns the optimal action which the environment offers. The literature on learning automata is extensive. We refer the reader to a review paper by Narendra and Thathachar [9] and a recent excellent book by Lakshmirarahan [3] for a review of the various families of learning automata. The latter reference also discusses in fair detail some of the applications of learning automata which include game playing [5], pattern recognition and hypothesis testing [9], priority assignment in a queueing system [7] and telephone routing [10,11].

Broadly speaking, learning automata can be classified into two categories: fixed structure automata, and Variable Structure Stochastic Automata (VSSA). A fixed structure automaton is one whose transition and output functions are time invariant. Examples of such automata are the Tsetlin, Krylov and Krinsky automata [17,18]. By far, most of the research in this area has involved the second category, namely, Variable Structure Stochastic Automata (VSSA). Automata in this category possess transition and output functions which evolve as the learning process proceeds. It can be shown that a VSSA is completely defined by a set of action probability updating functions [8,9,20].

VSSA are implemented using a Random Number Generator (RNG). The automaton decides on the action to be chosen based on an action probability

distribution. Nearly all the VSSA discussed in the literature permit probabilities which can take any value in the interval [0,1]. Hence the RNG required must theoretically possess infinite accuracy. In practice, however, the probabilities are rounded off to a certain number of decimal places depending on the architecture of the machine that is used to implement the automaton.

Learning automata can also be broadly classified in terms of their Markovian representations. Generally speaking, learning automata are either ergodic [10,13,14,15,19] or possess absorbing barriers [6,9,12]. Automata in the former class converge with a distribution which is independent of the initial distribution of the action probabilities. This feature is desirable when interacting with a nonstationary environment - for the automaton does not "lock itself" into choosing any one action. However, if the environment is stationary an automaton with an absorbing barrier is preferred. Various absolutely expedient schemes which ideally interact with such environments have been proposed in the literature [3,6,8,9].

To minimize the requirements on the RNG and to increase the speed of convergence of the VSSA the concept of discretizing the probability space was recently introduced in the literature [12,16]. As in the continuous case, a discrete VSSA is defined using a probability updating function. However, as opposed to the functions used to define continuous VSSA, discrete VSSA utilize functions that can only assume a finite number of values. These values divide the interval [0,1] into a finite number of subintervals. If the subintervals are all of equal length the VSSA automaton is said to be linear. Using these functions discrete VSSA

can be designed - the learning being performed by updating the action probabilities in discrete steps.

Various experimental results involving discretized Reward-Inaction automata were first reported by Thathachar and Oommen [16]. The first theoretical results concerning discretized Automata were proved in [12]. The latter paper concerned the ε-optimality of the two-action discretized Linear Reward-Inaction automaton. In this paper we shall deal with various discretized automata and prove their asymptotic properties. In particular we shall show that for the two action case:

(i) The Discretized Linear Reward-Inaction (DL_{RI}) automaton is absorbing and ε-optimal in all random environments.

(ii) The Discretized Linear Inaction-Penalty (DL_{IP}) automaton is ergodic and expedient in all random environments.

(iii) The Discretized Linear Inaction-Penalty automaton with artificially created absorbing barriers is ε-optimal in <u>all</u> random environments. This is the only scheme known to us which is of an inaction-penalty flavour and which is simultaneously ε-optimal.

(iv) The family of Discretized Nonlinear Reward-Inaction (DN_{RI}) automata is ε-optimal in all random environments. Further, we shall derive the maximum advantage that can be obtained by nonlinearizing the automaton.

The results concerning Discretized Linear Reward-Inaction (DL_{RI}) automata are primarily available in the literature [12,16]. Only those results in [12] which are required to derive the properties of discretized inaction-penalty and Discretized Nonlinear Reward-Inaction (DN_{RI}) automata

will be proved here. For the sake of brevity, the rest of the theoretical results concerning the DL_{RI} scheme will be merely alluded to and not reiterated. However, for the sake of continuity and for the purpose of this paper serving as a comprehensive report on the state of knowledge concerning discretized automata, the unproven results will at least be stated.

Apart from the above theoretical results various simultation results concerning Discretized Linear Reward-Penalty (DL_{RP}) automaton will also be presented. The latter family is ergodic. Based on the simulation results we conjecture that this family is ε-optimal. If this conjecture is true, this automaton will surely prove to be a very powerful learning machine - overcoming the major drawback of its continuous counterpart - namely the fact that the continuous L_{RP} automaton with equal coefficients is never ε-optimal but is at its best only expedient. Besides, the DL_{RP} automaton possesses excellent properties when interacting with non-stationary environments.

Various open problems concerning discretized automata will be posed in the concluding section of this paper.

We shall first present some fundamentals and introduce the notation we shall use. The DL_{RI} scheme will then be studied. Subsequently, the DL_{IP} automaton and its absorbing generalization will be analyzed. We then consider nonlinear schemes and shall show the ε-optimality of the DN_{RI} scheme. We shall conclude the paper by discussing the DL_{RP} automaton and conjecturizing its ε-optimality.

I.1 Fundamentals

The automaton considered in this paper (Figure 1) selects an action $a(n)$ at each instant 'n' from a finite action set $\{a_i | i = 1 \text{ to } R\}$. The selection is done on the basis of a probability distribution $\underline{p}(n)$, an $R \times 1$ vector where,

$$p_i(n) = Pr[a(n) = a_i],$$

$$\sum_{i=1}^{R} p_i(n) = 1 \text{ for all } n. \qquad (1)$$

The selected action serves as the input to the environment which gives out a response $b(n)$ at time 'n'. $b(n)$ is an element of $B = \{0,1\}$. The response '1' is said to be a 'penalty'. The environment penalizes the automaton with the penalty c_i, where,

$$c_i = Pr[b(n) = 1 | a(n) = a_i] \qquad (i = 1 \text{ to } R). \qquad (2)$$

Thus the environment characteristics are specified by the set of penalty probabilities $\{c_i\}$ $(i = 1 \text{ to } R)$. On the basis of the response $b(n)$ the action probability vector $\underline{p}(n)$ is updated and a new action chosen at $(n+1)$.

The reward probability d_i is defined as $1-c_i$ for $i = 1$ to R.

The $\{c_i\}$ are unknown initially and it is desired that as a result of interaction with the environment the automaton arrives at the action which evokes the minimum penalty response in an expected sense. It may be noted that if

$$c_L = \min_i (c_i) \qquad (3)$$

then $p_L(n) = 1$, $p_i(n) = 0$ for $i \neq L$ achieves this result. Updating schemes for $\underline{p}(n)$ are to be chosen with this optimal solution in view. Throughout this paper we deal with the case when R, the numbers of actions, is two.

I.2 Learning Criteria

With no apriori information, the automaton chooses the actions with equal probability. The expected penalty is thus initially M_0, the mean of the penalty probabilities.

An automaton is said to learn expediently if, as time tends towards infinity, the expected penalty is less than M_0. We denote the expected penalty at time 'n' as $E[M(n)]$. The automaton is said to be optimal if $E[M(n)]$ equals the minimum penalty probability in the limit as time goes towards infinity.

It is ε-optimal if in the limit $E[M(n)] < c_{min} + \varepsilon$ where $c_{min} = \min\{c_i\}$, for any arbitrary $\varepsilon > 0$ by suitable choice of some parameter of the automaton. Thus the limiting value of $E[M(n)]$ can be as close to c_{min} as desired.

II. THE DISCRETIZED LINEAR REWARD-INACTION (DL_{RI}) AUTOMATON

The Discretized Linear Reward-Inaction (DL_{RI}) automaton has $(N + 1)$ states where N is an even integer. We refer to the set of states as $S = \{s_0, s_1, \ldots, s_N\}$. Associated with the state s_i is the probability $\frac{i}{N}$, and this represents the probability of the automaton choosing action a_1. Note that in this state the automaton chooses the action a_2 with probability $(1 - \frac{i}{N})$. Since any one of the action probabilities completely defines the vector of action probabilities, we shall, with no loss of generality, consider $p_1(n)$.

The basic idea in the learning process is to make discrete changes in the action probabilities. By defining the transition map as a function from S X B to S the changes in the action probabilities are indeed discrete. For the DL_{RI} automaton the transition map is defined by (4) below for $s(n) = s_i \neq s_0$ or s_N.

$$\begin{aligned} s(n+1) &= s_{i+1} && \text{if } a(n) = a_1 \text{ and } b(n) = 0, \\ &= s_{i-1} && \text{if } a(n) = a_2 \text{ and } b(n) = 0, \\ &= s_i && \text{if } a(n) = a_1 \text{ or } a_2 \text{ and } b(n) = 1. \end{aligned} \tag{4}$$

s_0 and s_N are absorbing states and hence if $s(n) = s_0$ then $s(n+1) = s_0$ and if $s(n) = s_N$, then $s(n+1) = s_N$, for all n. The state transitions are shown in Figure 2.

Observe that, by virtue of the probabilities associated with the states, the updating can alternatively be described in terms of the

action probabilities as follows: If $p_1(n) \neq 0$ or 1,

$$\begin{aligned} p_1(n+1) &= p_1(n) + \frac{1}{N} && \text{if } a(n) = a_1,\ b(n) = 0, \\ &= p_1(n) - \frac{1}{N} && \text{if } a(n) = a_2,\ b(n) = 0, \\ &= p_1(n) && \text{if } a(n) = a_1 \text{ or } a_2,\ b(n) = 1. \\ p_1(n+1) &= p_1(n) && \text{if } p_1(n) = 0 \text{ or } 1. \end{aligned} \tag{5}$$

The automaton does not update the action probabilities when there is a penalty response from the environment and hence is called a reward-inaction automaton [4]. Further since it is discretized and linear we shall refer to it as the DL_{RI} automaton.

II.1 Markovian Representation of the DL_{RI} Automaton

It is evident from the previous description that $\{p_1(n)\}$ behaves as a homogeneous Markov chain with two absorbing states. Furthermore it is a random walk with the transition probabilities dependent on the state occupied. If $d_1 = 1 - c_1$ and $d_2 = 1 - c_2$, the transition probabilities are as follows:

Let $P_{i,j} = \Pr[a(n+1) = a_j \mid a(n) = a_i]$. Then,

$$\begin{aligned} P_{i,i+1} &= g_i d_1, \\ P_{i,i-1} &= \bar{g}_i d_2, \\ P_{i,i} &= 1 - g_i d_1 - \bar{g}_i d_2, \end{aligned}$$

where $i \neq 0, N$ and $g_i = \frac{i}{N}$, $\bar{g}_i = 1 - \frac{i}{N}$. (6)

Besides $P_{0,0} = 1$ and $P_{N,N} = 1$. All other transition probabilities are zero.

As s_0 and s_N are absorbing states and there are no periodic states, it is clear that $p_1(n)$ tends to 0 or 1 with probability 1 [2]. However, only one of these terminal values is desired. For fixing ideas let us assume $c_1 < c_2$. Then it is desired that $p_1(n)$ should tend to 1 and $s(n)$ to s_N.

We shall prove that this is indeed the case.

II.2 Asymptotic Optimality of the DL_{RI} Scheme

With no loss of generality we assume that $p_1(0) = p_2(0) = 0.5$. Since N is even this implies that the initial state of automaton is $s_{N/2}$. Further, as stated earlier, we assume that $c_1 < c_2$. Thus, we intend to prove that if a N-state DL_{RI} automaton is used to perform the learning,

$$\lim_{N\to\infty} \lim_{n\to\infty} p_1(n) = 1.$$

Definition: X_k is the random variable which takes the value i if the DL_{RI} automaton is in state s_i at the kth time instant. X_k is merely the index of the state of the automaton at the kth time instant.

Let $f_i = Pr[a(n+1) = a_{i+1} | a(n) = a_i]$ and $u_i = Pr[a(n+1) = a_{i-1} | a(n) = a_i]$

Then, from (6), if $g_i = \frac{i}{N}$,

$$f_i = g_i d_1 \tag{7}$$

and

$$u_i = (1 - g_i)d_2. \tag{8}$$

Further, $\Pr[a(n+1) = a_i | a(n) = a_i] = 1 - f_i - u_i$. (9)

Note that $f_i u_i > 0$ for all $i = 1, \ldots, N-1$ and that since s_0 and s_N are absorbing states, $f_o = u_o = f_N = u_N = 0$.

Theorem I.

Let $R_o = 1$ and $R_i = \sum_{j=1}^{i} \frac{u_j}{f_j}$. Further let

$$Z_k = \sum_{i=0}^{X_k - 1} R_i \,. \tag{10}$$

The quantity Z_k is a martingale.

Proof.

Alluding to the Markov property, observe that it is sufficient to prove that

$$E[Z_{k+1} | X_k] = Z_k.$$

By definition if $X_k = 0$, $Z_k = 0$.

Now consider $E[Z_{k+1} | X_k]$. If $I_{\{\cdot\}}$ is the indicator function of the event in $\{\cdot\}$,

$$E[Z_{k+1} | X_k] = Z_k \cdot I_{\{X_k=0 \text{ or } X_k=N\}} + [f_{X_k} (Z_k + R_{X_k}) + u_{X_k} (Z_k - R_{X_k - 1})$$

$$+ (1 - f_{X_k} - u_{X_k}) Z_k] I_{\{0 < X_k < N\}}.$$

Consider the second term. The terms $f_{X_k} Z_k$ and $u_{X_k} Z_k$ cancel. Further,

$$f_{X_k} R_{X_k} - u_{X_k} R_{X_k-1} = f_{X_k} (R_{X_k} - \frac{u_{X_k}}{f_{X_k}} \cdot R_{X_k-1}) = 0.$$

Hence $E[Z_{k+1} | X_k] = 0$ for all k, and thus Z_k is a martingale. Due to the above theorem, the limiting value of the probability of the DL_{RI} scheme converging to the state s_N can be easily derived.

Theorem II.

For any even N, the probability of converging to s_N is given by

$$H_N = \left(\sum_{i=0}^{N/2-1} R_i \right) / \left(\sum_{i=0}^{N-1} R_i \right) \qquad (11)$$

where $R_i = \prod_{j=1}^{i} \frac{u_j}{f_j}$ and u_j and f_j are defined by (7) and (8) respectively.

Proof.

Since $Z_k = \sum_{i=0}^{X_k-1} R_i$ is a martingale

$$E[Z_{k+1}] = E[Z_k] \text{ for all } k.$$

Since at k=0, $X_k = N/2$, we observe that for all k, Z_k must have the value

$$Z_o = \sum_{i=0}^{N/2-1} R_i.$$

Let T be the time for absorption. Then,

$$E[Z_T] = E[Z_o] = \sum_{i=0}^{N/2-1} R_i.$$

Whence,

$$0 \cdot \Pr(X_T=0) + \left(\sum_{i=0}^{N-1} R_i\right) \cdot \Pr(X_T=N) = \sum_{i=0}^{N/2-1} R_i$$

which proves the theorem.

The above two theorems lead to the ε-optimality of the DL_{RI} scheme. They will also be used in the results concerning the ADL_{IP} and DN_{RI} automata.

Theorem III.

The DL_{RI} scheme is ε-optimal in all random environments.

Proof.

It is required to prove that

$$\lim_{N\to\infty} \lim_{n\to\infty} p_1(n) = 1.$$

For a given N, consider

$$R_i = \prod_{j=1}^{i} \frac{u_j}{f_j}$$

$$= \left(\prod_{j=1}^{i} \left(\frac{N-j}{N} \cdot d_2\right)\right) / \left(\prod_{j=1}^{i} \left(\frac{j}{N} d_1\right)\right)$$

$$= \prod_{j=1}^{i} \frac{N-j}{j} \left(\frac{d_2}{d_1}\right) = \binom{N-1}{i} \left(\frac{d_2}{d_1}\right)^i$$

Let $e = \frac{d_2}{d_1}$. Since we assume that a_1 is the desired action, $e < 1$.

Hence, by Theorem II, the probability of being absorbed in state N is

$$H_N = \left(\sum_{i=0}^{N/2-1} \binom{N-1}{i} e^i \right) / \left(\sum_{i=0}^{N-1} \binom{N-1}{i} e^i \right)$$

Indeed, by Theorem IV, $\lim_{N\to\infty} H_N = 1$, and the theorem is proved.

Theorem IV.

Let

$$H_N = \left(\sum_{i=0}^{N/2-1} \binom{N-1}{i} e^i \right) / \left(\sum_{i=0}^{N-1} \binom{N-1}{i} e^i \right)$$

Then, $\lim_{N\to\infty} H_N = 1$.

Proof.

The proof is quite involved and found in [12]. It is omitted for the sake of brevity.

Remark: (i) From the proof of Theorem IV it can be shown that H_N is exactly the ratio of $B_q(N/2, N/2)$ to $B(N/2, N/2)$, where $q = \frac{1}{1+e}$, and $B_q(\cdot,\cdot)$ and $B(\cdot,\cdot)$ are Incomplete Beta Function and the Beta Function respectively [12]. Thus, for a given N the final value of $E[p_1(\infty)]$ can be precomputed if the values of the penalty probabilities are known. Note that this does not invalidate the learning problem, because though the values of the penalty probabilities may be known the actions to which they correspond may be unknown.

(ii) A word regarding the design of the DL_{RI} automaton is not out of place. If the values of c_1 and c_2 are known (and the actions to which these values correspond are unknown), the designer is interested in

deciding on the value of N to be used to guarantee a tolerated expected number of erroneous decisions. This value of N can be obtained from tabulated values of the ratio of $B_q(\frac{N}{2}, \frac{N}{2})$ to $B(\frac{N}{2}, \frac{N}{2})$ [21]. In Table I below, we have given for typical values of N the maximum values of $\frac{d_2}{d_1}$ which will guarantee the specified degree of accuracy in $E[p_1(\infty)]$. Thus, if N=10, every environment with $\frac{d_2}{d_1} < 0.43055$ will yield a value of $E[p_1(\infty)]$ greater than or equal to 0.9.

TABLE I

N	$e_{0.75}$	$e_{0.9}$
10	0.64463	0.43055
20	0.73638	0.55746
30	0.77983	0.62248
40	0.80668	0.66417
60	0.83945	0.71674
120	0.88388	0.79077

Note: N: No. of subintervals in [0,1]

e_{acc}: Maximum value of $\frac{d_2}{d_1}$ to guarantee $E[p_1(\infty)] \geq$ acc.

Table indicating the minimum accuracy guaranteed by the DL_{RI} scheme.

Similarly, if $c_1 = 0.1$ and $c_2 = 0.8$, the ratio for $\frac{d_2}{d_1}$ has the value 0.22. Since for N=4, $e_{0.9}$ has the value 0.24347 (refer to page 256 of [21]), even an automaton with only 5 states will guarantee that $E[p_1(\infty)]$ will be greater than 0.9. Further, in this case, the mean-time for convergence is approximately only 4.1 units of time.

II.3 Mean Time For Absorption

A key factor in the evaluation of the automaton behaviour is the time taken to decide which action is the better one. This is well represented by the mean time taken for absorption and can be evaluated as follows. Let P_T be the submatrix of P which represents the transient states of the Markov chain, and let $B=[I-P_T]^{-1}$. Then, the Mean Time to Converge (MTC) from state s_i to either of the absorbing states is given by [2] as:

$$MTC(i) = \sum_{j=1}^{N-1} B_{ij}$$

where B_{ij} is the (i,j)th element of B. In particular, the mean time to converge from state $s_{\frac{N}{2}}$ is exactly

$$MTC(\frac{N}{2}) = \sum_{j=1}^{N-1} B_{\frac{N}{2} j} \qquad (12)$$

The computation of (12) is greatly simplified by utilizing the fact that B is tridiagonal. Indeed, it requires merly the computation of (N-1) elements of the $(\frac{N}{2})^{th}$ row of $[I-A_T]^{-1}$.

II.4 Numerical Results and Comparison With Other Finite State Automata

The analysis made so far has concentrated on two aspects of behaviour of the automaton. The probability of absorption to the desired state as given by $E[p_1(\infty)]$ represents the accuracy of operation and the mean time for absorption is a measure of the speed of convergence of the automaton. These two quantities are computed for several values of N, c_1 and c_2 and shown in Table II.

Comparisons with other automata on a common basis are hard to make, but to have an idea of the behaviour of deterministic automata under similar conditions, the Tsetlin and Krinsky automata with N states are considered. Such a comparison is subject to the limitation that states correspond to different entities in the DL_{RI} and the deterministic automata. The values of $p_1(\infty)$ for the deterministic automata are also shown in Table II. Similar information is provided by Figure 3 where $E[p_1(\infty)]$ is plotted w.r.t. c_1 when $c_2 = 0.8$ and $N = 10$.

TABLE II

c_1	c_2	N	DL_{RI}		Tsetlin	Krinsky
			$E[p_1(\infty)]$	Mean Time For Convergence (No. of Iterations)	$p_1(\infty)$	$p_1(\infty)$
0.4	0.8	4	0.84	6	0.80	0.80
		10	0.95	16	0.95	0.97
		20	0.99	31	0.99	0.99
0.6	0.8	4	0.74	9	0.64	0.64
		10	0.86	26	0.72	0.81
		20	0.94	53	0.75	0.95
0.7	0.8	4	0.65	11	0.57	0.57
		10	0.73	34	0.60	0.66
		20	0.81	76	0.60	0.79

Comparative Performance of DL_{RI}, Tsetlin and Krinsky Automata

Some results not shown in Table II relate to the speed of convergence of the Tsetlin and Krinsky automata. The number of iterations required for the action probabilities to reach within 0.01 of their final values were 30 and 16 for the case $c_1 = 0.4$, $c_2 = 0.8$, $N = 4$. These values

moved to 68 and 56 when N = 6. When the penalty probabilities became closer as when $c_1 = 0.7$ and $c_2 = 0.8$, the Tsetlin automaton needed 175 iterations and the Krinsky, 12 for the case of N = 4. For N = 10, the Tsetlin automaton required over 2000 iterations and the Krinsky about 50. The extreme slowness of the Tsetlin automaton in the latter case is possibly because the minimum penalty probability is greater than 0.5.

The following qualitative conclusions can be drawn from the results shown.

(1) For a small number of states ($N \approx 4$) and environments with a large difference in penalty probabilities, the DL_{RI} automaton is more accurate than the deterministic automata.

(2) For environments with $c_1 > 0.5$, the DL_{RI} is more accurate than the Tsetlin automaton for all N.

(3) In general, for a fixed N, as the difference between the penalty probabilities is decreased, DL_{RI} becomes more accurate than the Tsetlin and Krinsky automata.

(4) It was observed that in all the environments considered, the DL_{RI} automaton was faster than the Krinsky and Tsetlin automata.

II.5 Comparison of the DL_{RI} Automaton with the L_{RI} Scheme

For any environment (c_1, c_2) Lakshmivarahan and Thathachar [3] have shown that for the L_{RI} scheme, the lower bound for the accuracy of convergence is obtained by solving for x_i from

$$\frac{e^{-x_i\theta}}{-x_1\theta} = \frac{1 - c_i}{1 - c_j} \qquad i,j = 1,2 \; ; \; i \neq j,$$

where θ is the parameter of the L_{RI} scheme.
If this value of x_i is substituted in

$$\Gamma_L(p_i) = \frac{1-e^{-x_i p_i}}{1-e^{-x_i}} \tag{13}$$

we obtain a lower bound for converging to the action which penalty probability c_i, given that the initial probability of choosing a_i is p_i.

Using the above bound we have experimentally compared the L_{RI} and the DL_{RI} schemes. For example, for the environment $c_1 = 0.2$, $c_2 = 0.6$, the minimum accuracy that is obtained is 0.998, when $\theta = 0.1$ if the initial starting value is 0.5. Similarly, with $N = 70$, the lower bound for the accuracy obtained using the DL_{RI} scheme is also 0.998.

The two automata were now made to learn the best action from the above environment and the following results were obtained.

In 240 iterations the L_{RI} scheme gave only an expected value of 0.99982. The DL_{RI} scheme gave an expected value of 0.99999 and subsequently the value stayed at unity. This definitely indicates the superiority of the DL_{RI} scheme. If a stopping criterion was used, it was seen that $E[p_1(n)]$ reached 0.99 in 125 iterations with the DL_{RI} scheme. The L_{RI} automaton took 135 iterations to converge to the same limit. Thus on both counts (speed and accuracy) the DL_{RI} scheme seems to be superior to the L_{RI} scheme.

We now terminate our discussion on the DL_{RI} scheme and proceed to investigate the properties of Discretized Inaction-Penalty Automata.

III. DISCRETIZED LINEAR INACTION-PENALTY AUTOMATA

In this section we consider two discretized automata which are of an inaction-penalty flavour. The first machine (referred to as the DL_{IP} automaton) has been shown to be ergodic (i.e., non-absorbing) and expedient. We then consider a slight modification of the latter automaton - an inaction-penalty automaton which is rendered artificially absorbing. The latter modification is called the Absorbing DL_{IP} automaton (or ADL_{IP}). Amazingly enough, merely altering the end states to be absorbing completely transforms the scheme and renders it ε-optimal. The ADL_{IP} scheme is the only scheme known to us which is of an inaction-penalty nature and simultaneously ε-optimal.

It has been well known that the updating function of a learning automaton must be dependent on the response it receives from the environment. For example, consider a continuous VSSA which completely ignores the penalty responses of the environment. Such an automaton is of the Reward-Inaction type, and it is well known that there are Linear and nonlinear Reward-Inaction schemes which are both absolutely expedient and ε-optimal. Apart from the continuous schemes, indeed as shown in the previous section, even discretized ε-optimal schemes of the Reward-Inaction flavour do exist. It is in this connection that we believe that the introduction of the ADL_{IP} scheme is a major contribution. Although continuous inaction-penalty schemes are at their best expedient (and definitely not absolutely

By virtue of the probabilities associated with the states, the updating can alternatively be described in terms of the action probabilities as follows:

$$\begin{aligned} p_1(n+1) &= p_1(n) + \frac{1}{N} && \text{if } a(n) = a_2,\ b(n) = 1, \\ &= p_1(n) - \frac{1}{N} && \text{if } a(n) = a_1,\ b(n) = 1, \\ &= p_1(n) && \text{if } a(n) = a_1 \text{ or } a_2,\ b(n) = 0. \end{aligned} \tag{15}$$

As in the DL_{RI} scheme, since $p_2(n) = 1-p_1(n)$, (15) fully defines the changes made on both the action probabilities. The way the action probabilities are updated warrants the automatons name.

III.2 The Asymptotic Properties of the DL_{IP} Automaton

Let us consider the Markovian properties of the DL_{IP} automaton. $p_1(n)$ behaves as a homogeneous Markov chain defined by a stochastic matrix M. $M_{i,j}$, the arbitrary element of M is defined as:

$$M_{i,j} = Pr[s(n) = s_j | s(n-1) = s_i].$$

From (15), the elements of M can be written down as below:

$$\begin{aligned} M_{i,i-1} &= g_i c_1 && \text{for } 1 \leq i \leq N, \\ M_{i,i+1} &= \bar{g}_i c_2 && \text{for } 0 \leq i \leq N-1, \\ M_{i,i} &= 1 - g_i c_1 - \bar{g}_i c_2 && \text{for } 0 \leq i \leq N, \end{aligned} \tag{16}$$

where $g_i = \frac{i}{N}$ and $\bar{g}_i = 1 - \frac{i}{N}$. All the other elements of M are zero.

The Markov chain consists of exactly one closed communicating class. Further, since it is aperiodic the chain is ergodic and the limiting distribution is independent of the initial distribution [2]. Let $\underline{\pi}(n)$ be the state probability vector, where, for all n,

$$\underline{\pi}(n) = [\pi_0(n), \pi_1(n), \ldots, \pi_N(n)]^T, \text{ and,}$$

$$\pi_i(n) = \Pr[s(n) = s_i] \text{ with } \sum_{i=0}^{N} \pi_i(n) = 1.$$

Then the limiting value of $\underline{\pi}$ is given by $\underline{\pi}^*$ which satisfies,

$$M^T \underline{\pi}^* = \underline{\pi}^* \tag{17}$$

Using (17) we now derive the asymptotic properties of the DL_{IP} automaton.

Theorem V.

For the DL_{IP} Automaton whose memory depth is N, the limiting value of $\pi_k(n)$ is given by π_k^*, where,

$$\pi_k^* = \frac{\binom{N}{k} c_1^{N-k} c_2^{k}}{(c_1 + c_2)^N} .$$

Proof.

We intend to solve $M^T\underline{\pi}^* = \underline{\pi}^*$. We shall first obtain a solution $\underline{\pi}'$ where $M^T\underline{\pi}' = \underline{\pi}'$. Subsequently, we shall normalize $\underline{\pi}'$ to render it a probability vector. This normalized vector will indeed be $\underline{\pi}^*$. With no loss of generality, let $\pi_0' = c_1^N$. We inductively prove that $\pi_k' = \binom{N}{k} c_1^{N-k} c_2^{k}$.

Basic Step: Expanding the equation $M^T\underline{\pi}' = \underline{\pi}'$ for k=0 yields,

$$(1-c_2)\pi_0' + g_1 c_1 \pi_1' = \pi_0' .$$

Since $g_1 = \frac{1}{N}$, and $\pi_0' = c_1^N$, π_1' can be solved for. This yields,

$$\pi_1' = N \cdot \frac{c_2}{c_1} \cdot c_1^N = \binom{N}{1} c_1^{N-1} c_2.$$

Inductive Step: Assume that

$$\pi'_{k-1} = \binom{N}{k-1} c_1^{N-k+1} c_2^{k-1}$$

and

$$\pi'_k = \binom{N}{k} c_1^{N-k} c_2^{k}.$$

Expanding the equation in (17) which corresponds to π'_k, we get,

$$\bar{g}_{k-1} c_2 \pi'_{k-1} + (1-g_k c_1 - \bar{g}_k c_2) \pi'_k + g_{k+1} c_1 \pi'_{k+1} = \pi'_k.$$

Solving for π'_{k+1} yields,

$$\pi'_{k+1} = \frac{(g_k c_1 + \bar{g}_k c_2) \pi'_k - \bar{g}_{k-1} c_2 \pi'_{k-1}}{g_{k+1} c_1} . \qquad (18)$$

Substituting the inductive hypothesis, and noting that for all i, $g_i = \frac{i}{N}$ and $\bar{g}_i = \frac{N-i}{N}$, (18) can be rewritten as:

$$\pi'_{k+1} = \frac{[\frac{k}{N} c_1 - \frac{N-k}{N} c_2] \binom{N}{k} c_1^{N-k} c_2^{k} - [\frac{N-k+1}{N} c_2] \binom{N}{k-1} c_1^{N-k+1} c_2^{k-1}}{\frac{k+1}{N} c_2}$$

which after considerable algebra simplifies to

$$\pi'_{k+1} = \binom{N}{k+1} c_1^{N-k-1} c_2^{k+1} .$$

Thus for all i,

$$\pi'_i = \binom{N}{i} c_1^{N-i} c_2^{i} .$$

Normalizing to get π^*_i yields,

$$\pi^*_i = \frac{\pi'_i}{\sum_{i=0}^{N} \pi'_i} = \frac{\binom{N}{i} c_1^{N-i} c_2^{i}}{\sum_{j=0}^{N} \binom{N}{j} c_1^{N-j} c_2^{j}} = \frac{\binom{N}{i} c_1^{N-i} c_2^{i}}{(c_1 + c_2)^N} .$$

We now derive the limiting values of $E[p_1(n)]$ and $Var[p_1(n)]$.

Theorem VI

The limiting distribution of $p_1(n)$ has the following mean and variance:

$$E[p_1(\infty)] = \frac{c_2}{c_1+c_2}$$

$$\text{Var}\,[p_1(\infty)] = \frac{c_1\, c_2}{N(c_1+c_2)^2} \cdot$$

Proof: From theorem I,

$$\pi_k^* = \binom{N}{k} \frac{c_1^{N-k}\, c_2^{k}}{(c_1+c_2)^N}$$

By regrouping the terms, we can equivalently write,

$$\pi_k^* = \binom{N}{k} \left(\frac{c_1}{c_1+c_2}\right)^{N-k} \left(\frac{c_2}{c_1+c_2}\right)^{k}$$

$$= \binom{N}{k} q^k (1-q)^{N-k}, \quad \text{where } q = \frac{c_2}{c_1+c_2} \cdot$$

Let X^* be the limiting index of the state of the automaton. From the above, clearly X^* is Binomially distributed, with parameters N and q. Hence,

$$E[X^*] = Nq$$

and

$$\text{Var}\,[X^*] = N\, q(1-q)$$

Thus, $E[p_1(\infty)] = \frac{1}{N} E[X^*] = q = \frac{c_2}{c_1+c_2}$ and $\text{Var}[p_1(\infty)] = \frac{1}{N^2} \text{Var}[X^*] =$

$\frac{1}{N} q(1-q) = \frac{c_1\, c_2}{N(c_1+c_2)^2}$ and the theorem is proved.

Corollary VI.I

Independent of the number of states it possesses the DL_{IP} automaton is expedient in all random environments.

Proof: We know that $E[M(n)] = c_1 E[p_1(n)] + c_2 E[p_2(n)]$. Using the results of Theorem II,

$$\lim_{n\to\infty} E[M(n)] = \frac{2 c_1 c_2}{(c_1+c_2)} .$$

The result follows since the RHS of the above equation is strictly less than $\frac{c_1 + c_2}{2}$.

III.2.1 Remarks

1. It is interesting to study the expressions for the mean and the variance of $p_1(\infty)$. From Theorem VII we observe that the mean is $\frac{c_2}{c_1+c_2}$ independent of N, the depth of memory of the machine. Further, the limiting variance decreases monotonically with N. Thus as M tends to infinity $p_1(n)$ tends to a random variable with mean $\frac{c_2}{c_1+c_2}$ and variance zero. In other words, $p_1(n)$ converges (with respect to N) in the mean square sense to the constant $\frac{c_2}{c_1+c_2}$.

2. Many expedient schemes converge with a value of $E[p_1(\infty)]=\frac{c_2}{c_1+c_2}$. Among these are the 2-state Tsetlin automaton [17,18], the symmetric Linear Reward-Penalty Scheme [3,8,9] and all the schemes which possess ergodicity of the mean [15]. Amazingly enough the DL_{IP} scheme has a mean action probability which converges to the same value.

3. Clearly the DL_{IP} automaton is neither ε-optimal nor absolutely expedient. We shall now modify the automaton and render it absorbing. The new automaton which results is ε-optimal.

III.3 The ADL_{IP} Automaton

The Absorbing Discretized Linear Inaction-Penalty (ADL_{IP}) automaton is obtained by defining the states s_0 and s_N of the DL_{IP} automaton to be absorbing. The automaton is formally defined as a pair (S,G) where,

(i) S is the set of states and is identical to the set of states of the DL_{IP} automaton, and,

(ii) G is the state transition map specified by (19) below for $1 \leq i \leq N-1$.

$$\begin{aligned} s(n+1) &= s_{i+1} && \text{if } a(n) = a_2 \text{ and } b(n) = 1, \\ &= s_{i-1} && \text{if } a(n) = a_1 \text{ and } b(n) = 1, \\ &= s_i && \text{if } a(n) = a_1 \text{ or } a_2 \text{ and } b(n) = 0. \end{aligned} \tag{19}$$

Further, s_0 and s_N are absorbing states, and thus, if $s(n) = s_0$ then $s(n+1) = s_0$, and if $s(n) = s_N$, then $s(n+1) = s_N$, for all N.

Obviously, $p_1(n)$ behaves as a homogeneous Markov chain with two absorbing states. Furthermore, it is a random walk with transition probabilities dependent on the state of the machine. Let Q be the

stochastic matrix defining the Markov chain, where the individual element of Q is,

$$Q_{i,j} = \Pr[s(n) = s_j | s(n-1) = s_i].$$

From the definition of the ADL_{IP} automaton, we write for $1 \leq i \leq N-1$,

$$\begin{aligned} Q_{i,i+1} &= \bar{g}_i \, c_2, \\ Q_{i,i-1} &= g_i \, c_1, \\ Q_{i,i} &= 1 - g_i \, c_1 - \bar{g}_i \, c_2, \end{aligned} \tag{20}$$

where $g_i = \frac{i}{N}$ and $\bar{g}_i = 1 - \frac{i}{N}$.

Besides $Q_{0,0} = Q_{N,N} = 1$. All the other elements of Q are zero. We now prove the asymptotic properties of the ADL_{IP} Scheme.

Theorem VII.

The ADL_{IP} automaton is ε-optimal in all random environments.

Proof.

It is required to prove that

$$\lim_{N \to \infty} \lim_{n \to \infty} p_1(n) = 1.$$

Using the arguments of Theorems I and II, we can write that for any even N, the probability of converging to s_N is given by:

$$H_N = \left(\sum_{i=0}^{N/2-1} R_i \right) / \left(\sum_{i=0}^{N-1} R_i \right)$$

In the case of the ADL_{IP} automaton,

$$R_i = \sum_{j=1}^{i} \frac{u_j}{f_j}, \text{ where}$$

$$u_j = \frac{j}{N} c_1 \text{ and } f_j = \left(\frac{N-j}{N}\right) c_2.$$

Simplifying the expression for R_i yields,

$$R_i = \prod_{j=1}^{i} \frac{u_j}{f_j} = \left(\prod_{j=1}^{i} \frac{j}{N} c_1\right) / \left(\prod_{j=1}^{i} \frac{N-j}{N} c_2\right)$$

$$= \prod_{j=1}^{i} \frac{j}{N-j} \frac{c_1}{c_2} = \frac{1}{\binom{N-1}{i}} \left(\frac{c_1}{c_2}\right)^i \tag{21}$$

Let $e = \frac{c_1}{c_2}$. Clearly $e < 1$, and hence the probability of being absorbed in state s_N is

$$H(N) = \left(\sum_{i=0}^{N/2-1} \frac{1}{\binom{N-1}{i}} e^i\right) / \left(\sum_{i=0}^{N-1} \frac{1}{\binom{N-1}{i}} e^i\right) \tag{22}$$

The result follows since by Theorem VIII H(N) tends to unity as $N \to \infty$.

Theorem VIII.

For all $0 < e < 1$,

$$\lim_{N \to \infty} H(N) = 1$$

where H(N) is given by (22). The theorem is proved in the Appendix.

III.4 Experimental Results

To evaluate the performance of the ADL_{IP} automaton, the latter was simulated and made to interact with various stationary environments whose penalty probabilities are (c_1, c_2). The various environments were obtained by varying c_1 from 0.1 to 0.7, while c_2 was kept constant at 0.8. The automaton interacted with each environment for 400 experiments so that a relatively accurate measure of the average performance of the automaton could be obtained.

The learning properties of the ADL_{IP} automaton was also compared with two other finite state learning machines - the 2N-state Tsetling automaton and the corresponding (N+1) State Discretized Linear Reward-Inaction (DL_{RI}) automaton. Figures 5 and 6 show the variation of $E[p_1(\infty)]$ with c_1, for the depths of memory of the machines being 4 and 10 respectively. Observe the superiority of the ADL_{IP} automaton in all environments for N=10, (Figure 6).

The ADL_{IP} automaton has one major drawback. It is rather slow in its convergence. The reason for this is probably because the end states are "artificially" rendered absorbing and so, if the machine is in an interior state it would behave <u>exactly</u> like the DL_{IP} automaton. To compare the DL_{RI} and the ADL_{IP} schemes we have given in Table III some typical values of the accuracy and the mean time to converge of the automata. Clearly the DL_{RI} scheme is to be chosen in a practical system if speed is the criterion.

TABLE III

		DL_{RI} Scheme		ADL_{IP} Scheme	
	c_1	$E[p_1(\infty)]$	M.T.C.	$E[p_1(\infty)]$	M.T.C.
N=4	0.2	0.896	4.61	0.960	9.87
	0.4	0.843	6.04	0.825	10.65
	0.6	0.741	8.52	0.655	10.57
N=10	0.2	0.980	11.46	1.00	70.28
	0.4	0.951	16.15	1.00	196.78
	0.6	0.855	25.58	0.93	499.11

A Comparison of the DL_{RI} and the ADL_{IP} Automata. In all the experiments $c_2 = 0.8$.

We now consider the effect of nonlinearizing the intervals in probability space.

IV. THE DISCRETIZED NONLINEAR REWARD-INACTION (DN_{RI}) AUTOMATON

One of the drawbacks of the DL_{RI} automaton is that is still belongs to a one-parameter family of automata. Once N is specified the value of $E[p_1(\infty)]$ and the speed of convergence are fixed in any given environment. More flexibility can be obtained by making the discrete values of the probabilities correspond to a nonlinear function of the states. In other words, g_j is a nonlinear function of j.

As in the linear case s_0 and s_N correspond to zero and unity respectively. Further, since the automaton in the state s_k chooses a_1 with probability g_k and a_2 with probability $1-g_k$, it can be seen that

$$g_k = 1 - g_{N-k} \qquad k = 0,1,\ldots,N.$$

Thus g_k need be specified only for values of k between 0 and $\frac{N}{2}$. Observe then that $g_{\frac{N}{2}}$ must be 0.5. For the DN_{RI} automaton, g_k is specified as:

$$g_k = 0.5\ (1 + (\frac{2k}{N} - 1)^{2J+1}) \qquad \text{for } 0 \leq k \leq \frac{N}{2}. \tag{23}$$

J is called the index of nonlinearity. J=0 corresponding to the linear case, and for the sake of simplicity J will be assumed to be integral. Observe that for all J, g satisfies the above boundary constraints.

A plot of $g(\cdot)$ as a function of the index of nonlinearity, J, is given in Figure 7, we now consider the convergence properties of the DN_{RI} scheme.

Theorem IX.

As $J \to \infty$, g_k tends to have the following form:

$$g_k = 0.5 \qquad 1 \leq k \leq N-1$$

$$g_0 = 0 \qquad \text{and} \qquad g_N = 1.0$$

Proof.

From (23) we observe that for all $1 \leq k < \frac{N}{2}$

$$\lim_{J \to \infty} g_k = \lim_{J \to \infty} 0.5 \left(1 - (x_k)^{2J-1}\right)$$

The result follows since $x_k > 0$ and $|x_k| = 1 - \frac{2k}{N} < 1$.

Theorem X.

For a fixed depth of memory N, the value of the probability of converging to a_1 tends to p_1^* as J tends to infinity, where,

$$p_1^* = \frac{d_1^{N/2}}{d_1^{N/2} + d_2^{N/2}}$$

Proof.

Using the results of Theorems I and II, we write,

$$p_1^* = \left(\sum_{i=0}^{N/2-1} R_i\right) / \left(\sum_{i=0}^{N-1} R_i\right)$$

where $R_i = \sum_{j=1}^{i} \frac{u_j}{f_j}$.

Note that in this case, by virtue of Theorem IX, as $J \to \infty$, $u_j \to \frac{d_2}{2}$ and $f_j \to \frac{d_1}{2}$. Hence,

$$R_i = \prod_{j=1}^{i} \frac{d_2}{d_1} = \left(\frac{d_2}{d_1}\right)^i .$$

Assume again, with no loss of generality, that a_1 is the superior action. Then, if $e = \frac{d_2}{d_1}$,

$$p_1^* = \left(\sum_{i=0}^{N/2-1} e^i\right) / \left(\sum_{i=0}^{N-1} e^i\right) = \left(\sum_{i=0}^{N/2-1} e^i\right) / \left(\sum_{i=0}^{N/2-1} e^i\right)\left(1 + e^{N/2}\right)$$

$$= \frac{1}{1 + \left(\frac{d_2}{d_1}\right)^{N/2}} = \frac{d_1^{N/2}}{d_1^{N/2} + d_2^{N/2}}$$

and the theorem is proved.

Theorem XI.

The DN_{RI} scheme is ε-optimal in all random environments.

Proof.

The result is quite straightforward from the previous theorem since,

$$\lim_{N\to\infty} p_1^* = 1.$$

IV.1 The DN_{RI} Scheme and Finite Memory Learning

Cover and Hellman [22] proved a powerful result concerning finite memory learning. Their result, which was initially posed in the Hypothesis Testing framework can easily be generalized to learning systems and be summarized as below:

(i) A learning machine with finite memory can never learn with accuracy that is arbitrarily close to unity.

(ii) If a learning machine has a memory capability of N, the maximum accuracy of learning that one can obtain from the machine is

$$\gamma_N = \frac{(c_2 d_1)^{\frac{N-1}{2}}}{(c_2 d_1)^{\frac{N-1}{2}} + (c_1 d_2)^{\frac{N-1}{2}}}$$

Consider the limiting case when $(c_1, c_2) = (0.5-\varepsilon, 0.5+\varepsilon)$, where ε is arbitrarily small. Notice too that this is possibly the most difficult environment for the learning machine to interact with, since c_1 and c_2 are almost equal. Furthermore, both are almost equal to 0.5. However, in this case, it is interesting to note that the DN_{RI} automaton attains this maximum accuracy as the index of nonlinearity, J, is increased indefinitely. Simulation results however, indicate that often a value of J even as small as 7 yields an accuracy comparable to that specified by Theorem X.

IV.2 Experimental Results

The DN_{RI} automaton with values of 'J' ranging from 3 to 7 was considered. The automaton was assumed to operate in environments with $c_2 = 0.8$ and different values of c_1. The values of $E[p_1(\infty)]$ and the mean time for absorption were computed averaged over 400 experiments. The results are shown in Table IV. The power of the DN_{RI} is obvious. For example, if $c_1 = 0.4$ and $c_2 = 0.8$, an 11 state DN_{RI} automaton with J=3 yielded an accuracy of 99.5% - the mean time for convergence being only 28 time units.

The results of the simulations are summarized below:

TABLE IV

c_1	N	J	$E[p_1(\infty)]$	Mean Time To Convergence
0.1	4	3	0.9400	5.3725
		7	0.9524	5.4950
	10	3	0.9975	15.6800
		7	1.0000	14.9275
0.4	4	3	0.8825	8.3900
		7	0.8825	8.3175
	10	3	0.9950	28.0250
		7	0.9975	26.2750
0.7	4	3	0.6500	15.4475
		7	0.6575	15.6375
	10	3	0.8350	84.7750
		7	0.8500	76.9750

Performance of the DN_{RI} automaton averaged over 400 experiments. In all cases $c_2 = 0.8$.

(i) Generally speaking, $E[p_1(\infty)]$ increase with increase in J in a given environment. However, generally speaking, increase in J decreases the speed of convergence. Thus, the introduction of the nonlinear function together with N provides means of controlling accuracy and speed simultaneously.

(ii) It may also be noted that maximum accuracy is obtained when J is large; i.e., probabilities are nearly constant for all the 'middle'

Observe that the machine is in state s_0 it has to choose a_2 and similarly if it is in s_N - it has to choose a_1. Thus the change in action probabilities can be written for $0 < p_1(n) < 1$ as:

$$\begin{aligned} p_1(n+1) &= p_1(n) + \frac{1}{N} && \text{if } a_1 \text{ is chosen and } b(n) = 0 \\ & && \text{or } a_2 \text{ is chosen and } b(n) = 1 \\ &= p_1(n) - \frac{1}{N} && \text{if } a_1 \text{ is chosen and } b(n) = 1 \\ & && \text{or } a_2 \text{ is chosen and } b(n) = 0 \end{aligned} \tag{25}$$

At the end states the following equality holds:

$$\begin{aligned} p_1(n+1) &= p_1(n) && \text{if } p_1(n) = 0 \text{ or } 1 \text{ and } b(n) = 0 \\ &= \frac{1}{N} && \text{if } p_1(n) = 0 \quad \text{and } b(n) = 1 \\ &= 1 - \frac{1}{N} && \text{if } p_1(n) = 1 \quad \text{and } b(n) = 1. \end{aligned}$$

If $c_1 < c_2$, the automaton has no absorbing barriers except in the degenerate case when $c_1 = 0$. This implies that the Markov chain is ergodic and that the limiting distribution of being in any state is independent of the corresponding initial distribution [2]. However, the limiting distribution is currently unknown. The problem of solving for it in the general case is by no means trivial. This is because the problem reduces to solving difference equations with coefficients that depend on the index of the state. However, for small values of N, the difference equation has been solved and a summary of the results is given below:

(i) For N=2, $E[p_1(\infty)] = \dfrac{c_2(1+c_2)}{c_1(1+c_1)+c_2(1+c_2)}$ (26)

(ii) The DL_{RP} scheme with N=2 is superior to the 2-state Tsetlin automaton in all random environments.

(iii) In all random environments the DL_{RP} scheme with N=2 is superior to the symmetric L_{RP} scheme and superior to all schemes which are ergodic in the mean [13,15].

V.1 Experimental Results and a Powerful Conjecture

The DL_{RP} automaton was simulated and its interaction with various environments in which c_2=0.8 was studied. c_1 was varied from 0.1 to 0.7. The learning capability of the machine as a function of number of states which it possessed has been tabulated in Table V.

TABLE V

c_1	$E[p_1(\infty)]$	$Var[p_1(\infty)]$
0.1	0.99897	0.00001
0.2	0.99654	0.00007
0.3	0.99249	0.00018
0.4	0.98196	0.00027
0.5	0.95193	0.00335
0.6	0.74069	0.00574

Variation of $E[p_1(\infty)]$ and $Var[p_1(\infty)]$ with the penalty probability c_1. In all cases N=100 and c_2 = 0.8

The typical variation of $E[p_1(\infty)]$ and $Var[p_1(\infty)]$ with N is shown in Figure 7 for the case when c_1 = 0.4 and c_2 = 0.8. Based on these and various other results, we propose the following conjecture.

Conjecture I.

The DL_{RP} scheme is ε-optimal in all random environments.

We now discuss the power of the DL_{RP} automaton when interacting with non-stationary environments.

V.2 The DL_{RP} Automaton in Non-Stationary Environments

Tsetlin who initiated work in Learning Automata did some work on the behaviour of his Automaton $L_{2N,2}$ in a Non-stationary environment [17,18].

The Automaton was made to switch between two environments E_1 and E_2 according to a Markov chain that determined the probability with which it was in either environment. If the probabilities of being in the ith environment at any time was given $p_{E_i}(n)$, then the probability of being in the same environment at the next instant was given by:

$$(1-\delta)p_{E_i}(n) + \delta p_{E_j}(n) \qquad i \neq j;\ i, j = 1,2. \qquad (27)$$

When δ is small, (27) states that with almost the same probabilities the same environment will be chosen in the next instant. A small value for δ thus implies a slowly varying Markov chain. The limiting value of the vector $[p_{E_1}, p_{E_2}]^T$ is $[0.5, 0.5]^T$, and so in the steady state, both the environments will be chosen with equal probabilities.

The mean time during which the automaton will be interacting with any particular environment can be easily shown to be $1/\delta$. If environment E_1 has penalty probabilities c_1 and $1-c_1$, in Tsetlin work, E_2 was so chosen to have penalty probabilities $1-c_1$ and c_1. The initial average penalty M_0 is thus 0.5

The expected value of the final penalty is compared to M_0 and the difference computed. Further, to reduce the errors incurred due to taking the sample mean as the expected value, the difference (M_0-M^*) was computed, where,

$$M^* = \frac{1}{K} \sum_{n=1}^{K} E[M(n)]$$

where K, the number of iterations done per run, was made very large. It is clear that a higher value of $(M_o - M^*)$ indicates a better automaton.

It was shown by Tsetlin that there was, for each environment, an optimal Memory for which $(M_o - M^*)$ was the Maximum. This memory was smaller for faster switching environment Markov chains (δ-large). Tsetlin's experiments proved that for faster switching environments, it was not advantageous to increase the memory. Storing the information regarding the previous environment chosen was not beneficial, if the mean time during which the particular environment interacted with the automaton, was small.

Theoretically (by considering a composite Markov chain), he proved that the $L_{2,2}$ was the best automaton, if

$$\frac{1-2}{\delta(1-\delta)} \leq \frac{1}{c(1-c)}$$

In such cases this is the best performance that any deterministic automaton can give (since Tsetlin L_{22} is equivalent to the $Krinsky_{22}$ automaton).

We have already indicated that the DL_{RP} scheme gives a higher accuracy (for all environments), than the L_{22} automaton. Thus in all environments where the L_{22} is the best deterministic automaton, the DL_{RP} will perform better, yielding a lower expected penalty and thus a higher value for $(M_o - M^*)$.

Given in Table VI are results that compare the performance of the

DL_{RP} automaton with the Tsetlin automaton and the conclusions drawn are explained thereafter.

TABLE VI

c/1-c	δ	DL_{RP}		Tsetlin	
		N	$(M_o - M^*)$	N_{opt}	$M_o - M^*$
0.45/0.55	0.45	2**	0.00376	2	0.001
		4	0.00194		
		6	0.00142		
		8	0.00149		
		10	0.00064		
0.45/0.55	0.10	2**	0.00600	4	0.005
		4	0.00509		
		6	0.00503		
0.45/0.55	0.01	2	0.00702		
		4	0.00893		
		6	0.00997		
		8	0.01298		
		10**	0.01338	14	0.017
		12	0.01255		

Note: (i) c/1-c indicates that E_1 has penalty probabilities (c,1-c) and E_2 has penalty probabilities (1-c,c).

(ii) N** indicates the optimum value of N for the DL_{RP} scheme and alongside it is entered the optimum memory for the L_{22} automaton with its (M_1-M^*).

Comparison of the DL_{RP} Scheme and $L_{2N,2}$ Automaton in Non-Stationary Environments.

The observation made on the DL_{RP} automaton learning in a non-stationary environment are:

(i) Only in environments for which the optimal memory is large (no exact limit has been derived) is the $L_{2N,2}$ superior to the DL_{RP}.

(ii) In many cases, even when the L_{22} is not the best automaton, but the memory is small, the DL_{RP} performs better than the $L_{2N,2}$ and with the advantage that the memory requirements is less (as is obvious from the third set of experiments conducted with the environment whose δ was 0.01 and c/1-c was 0.45/0.55 given in Table VI.

(iii) From Tsetlin's results [18] it is observed that for all environments which switch corresponding to a larger value of δ ($\delta > 0.32$), the optimal deterministic automaton is the L_{22}. Since the L_{22} always gives a higher expected penalty than the DL_{RP} we assert that in all such environments, the DL_{RP} will perform better and will give a lower value for $(M_o - M^*)$.

We thus conclude that in general, the DL_{RP} automaton performs better in most non-stationary environment at least for all ($0.32 \leq \delta \leq 1$). One must appreciate the fact that learning in a faster switching environment is more difficult than in a slower switching environment. This augmented with the fact that the penalty probabilities are close to each other makes the problem more difficult when both δ is small and the ratio c/1-c is near to unity. Simulation results show that in such environments, the $(M_o - M^*)$ obtained from the DL_{RP} is many times higher than the $(M_o - M^*)$ obtained by using the $L_{2N,2}$ automaton.

VI. CONCLUSIONS AND OPEN PROBLEMS

In this paper we have stated and proved asymptotic results concerning various variable structure stochastic automata. These automata however, unlike the automata discussed in the literature, change the

action probabilities in discrete jumps. The automata are called linear or nonlinear depending on whether or not these jumps are all of equal length. We have proved that the DL_{RI} and the DN_{RI} schemes are ε-optimal. The DL_{IP} scheme is shown to be ergodic and expedient. By artificially making the end states of the latter automaton absorbing, we have designed the ADL_{IP} automaton and proven its ε-optimality. We also have given simulation results for the Discretized Linear Reward-Penalty scheme and based on these we proposed a conjecture that the scheme is ε-optimal. The problem of proving or disproving this conjecture remains unsolved.

The problem of studying discretized nonlinear Inaction-Penalty and Reward-Penalty schemes is yet open. Finally, although simulation results exist [16] for multi-action discretized automata, the problem of theoretically analyzing their properties is still untackled.

ACKNOWLEDGEMENTS

I would like to thank Dr. E.R. Hansen of Lockheed Missiles and Space Co. Inc., for the proof of Theorem VIII. I am also grateful to a pioneer and giant in the area of learning automata, Professor M.A.L. Thathachar, of the Indian Institute of Science, Bangalore, India. His valuable comments and his encouragement during the course of this study have been much appreciated.

REFERENCES

[1] Flerov, Y.A., "Some Classes of Multi-Input Automata", Journal of Cybernetics", Vol.2, 1972, pp.112-122.

[2] Isaacson, D.L., and Madson, R.W., "Markov Chains: Theory and Applications", Wiley, 1976.

[3] Lakishmivarahan, S., "Learning Algorithms Theory and Applications", Springer-Verlag, New York, 1981.

[4] Lakishmivarahan, S., "ε-Optimal Learning Algorithms-Non-absorbing Barrier Type", Technical Report EECS 7901, February 1979, School of Electrical Engineering and Computing Sciences, University of Oklahoma, Norman, Oklahoma.

[5] Lakshmivarahan, S., "Two Person Decentralized Team With Incomplete Information", Applied Mathematics and Computation, Vol.8, pp.51-78, 1981.

[6] Lakshmivarahan, S., and Thathachar, M.A.L., "Absolutely Expedient Algorithms for Stochastic Automata", IEEE Trans. on Syst. Man and Cybern., Vol.SMC-3, 1973, pp.281-286.

[7] Meybodi, M.R., "Learning Automata and Its Application to Priority Assignment in a Queueing System With Unknown Characteristics", Ph.D. Thesis, School of Electrical Engineering and Computer Science, University of Oklahoma, Norman, Oklahoma.

[8] Narendra, K.S., and Thathachar, M.A.L., Forthcoming book on learning automata.

[9] Narendra, K.S., and Thathachar, M.A.L., "Learning Automata -- A Survey", IEEE Trans. Syst. Man and Cybern., Vol.SMC-4, 1974, pp.323-334.

[10] Narendra, K.S. and Thathachar, M.A.L., "On the Behaviour of a Learning Automaton in a Changing Environment With Routing Applications", IEEE Trans. on Syst. Man and Cybern., Vol.SMC-10, 1980, pp.262-269.

[11] Narendra, K.S., Wright, E. and Mason, L.G., "Application of Learning Automata to Telephone Traffic Routing", IEEE Trans. on Syst. Man and Cybern., Vol.SMC-7, 1977, pp.785-792.

[12] Oommen, B.J. and Hansen, E.R., "The Asymptotic Optimality of Discretized Linear Reward-Inaction Learning Automata", IEEE Trans. on Syst. Man and Cybern. May/June 1984, pp.542-545.

[13] Oommen, B.J., and Thathachar, M.A.L., "Multi-Action Learning Automata Possessing Ergodicity of the Mean", Proc. of the 1983 IASTED Symposium on Measurement and Control, MECO 83, pp.61-64.

[14] Poznyak, A.S., "Use of Learning Automata for the Control of Random Search", Automation and Remote Control, Vol.33, No.12, 1972, pp.1992-2000.

[15] Thathachar, M.A.L., and Oommen, B.J., "Learning Automata Possessing Ergodicity of the Mean: The Two Action Case", IEEE Trans. for Syst. Man and Cybern., Vol.SMC-13, 1984, pp.1143-1148.

[16] Thathachar, M.A.L., and Oommen, B.J., "Discretized Reward-Inaction Learning Automata", Journal of Cybernetics and Information Sciences, Spring 1979, pp.24-29.

[17] Tsetlin, M.L., "On the Behaviour of Finite Automata in Random Media", Automat. Telemekh., Vol.22, 1961, pp.1345-1354.

[18] Tsetlin, M.L., "Automaton Theory and the Modelling of Biological Systems", New York and London, Academic, 1973.

[19] Tsypkin, Y.Z. and Poznyak, A.S., "Finite Learning Automata", Engineering Cybernetics, Vol.10, 1972, pp.478-490.

[20] Varshavskii, V.I., and Vorontsova, I.P., "On the Behaviour of Stochastic Automata With Variable Structure", Automat. Telemek. (USSR), Vol.24, 1963, pp.327-333.

[21] CRC Handbook of Tables for Probability and Statistics, Published by the Chemical Rubber Company, Cleveland, Ohio, 2nd Edition, 1968.

[22] Hellman, M.E., and Cover, T.M., "Learning With Finite Memory", Annals of Mathematical Statistics, 1970, Vol.41, pp.765-782.

APPENDIX

Proof of Theorem VIII

Consider the sum

$$Q(N) = \sum_{K=0}^{N-1} \psi(K)$$

where $\psi(K) = e^K \theta(K)$ and

$$\theta(K) = \frac{1}{\binom{N-1}{K}}$$

We wish to determine

$$\lim_{N\to\infty} \frac{Q(N/2)}{Q(N)} .$$

Note that

$$\frac{\psi(K+1)}{\psi(K)} = \frac{(K+1)e}{N-1-K} ,$$

and that for $K \leq \frac{N}{2} - 1$,

$$\frac{\psi(K+1)}{\psi(K)} \leq e < 1 .$$

Thus $\psi(K)$ decreases monotonically with K. In particular,

$$\psi(K) < \psi(2) = \frac{2e^2}{(N-1)(N-2)}$$

for all $K=3, \ldots, \frac{N}{2} - 1$, so that

$$\sum_{K=2}^{\frac{N}{2}-1} \psi(K) < (\tfrac{N}{2} - 2) \frac{2e^2}{(N-1)(N-2)} = O(N^{-1}) .$$

Therefore,

$$Q(\tfrac{N}{2}) = \psi(0) + \psi(1) + O(N^{-1}).$$

Since $\psi(0) = 1$ and $\psi(1) = \frac{e}{N-1} = O(N^{-1})$, we have

$$Q(\tfrac{N}{2}) = 1 + O(N^{-1}). \tag{1}$$

In order to consider $Q(N)$, define

$$T(N) = \sum_{K=N/2}^{N-1} e^{K}\,\theta(K)$$

and note that $Q(N) = Q(\frac{N}{2}) + T(N)$. Since $0 \le e < 1$,

$$T(N) \le e^{N/2} \sum_{K=N/2}^{N-1} \theta(K) \tag{2}$$

Now

$$\frac{\theta(K+1)}{\theta(K)} = \frac{N+1}{N-1-K}$$

and for $K \ge N/2$,

$$\frac{\theta(K+1)}{\theta(K)} \ge \frac{\frac{N}{2}+1}{\frac{N}{2}-1} > 1.$$

Therefore, for $N/2 \le K \le N-4$,

$$\theta(K) < \theta(N-3) = \frac{2}{(N-1)(N-2)}$$

so that

$$\sum_{K=N/2}^{N-1} \theta(K) = \sum_{K=N/2}^{N-3} \theta(K) + \theta(N-2) + \theta(N-1)$$

$$< (\tfrac{N}{2}-2)\,\frac{2}{(N-1)(N-2)} + \frac{1}{N-1} + 1$$

$$= 1 + O(N^{-1}) \tag{3}$$

From (2) and (3),

$$T(N) \leq e^{N/2} [1 + O(N^{-1})] .$$

Since $0 \leq e < 1$,

$$T(N) = O(e^{N/2}). \tag{4}$$

Since

$$Q(N) = Q(N/2) + T(N),$$

(1) and (4) yield

$$Q(N) = 1 + O(N^{-1}) + O(e^{N/2})$$

$$= 1 + O(N^{-1}) .$$

Therefore, using (1)

$$\frac{Q(N/2)}{Q(N)} = \frac{1 + O(N^{-1})}{1 + O(N^{-1})} = 1 + O(N^{-1})$$

and

$$\lim_{N\to\infty} \frac{Q(N/2)}{Q(N)} = 1.$$

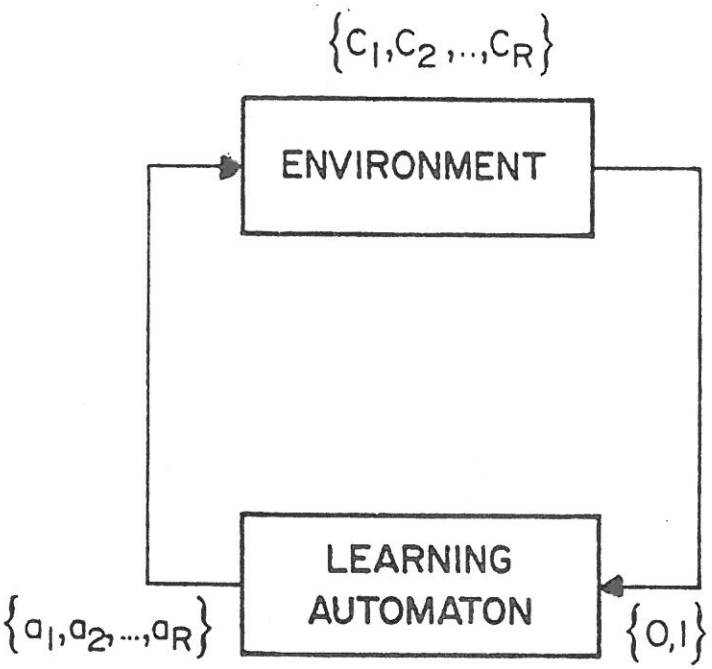
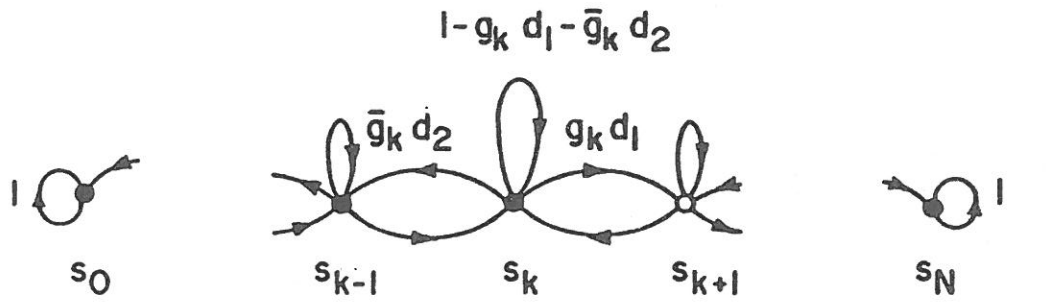


FIG.1: The Learning Automaton



N : EVEN

$g_k = k/N$

$\bar{g}_k = 1 - k/N$

FIG. 2 : The $DL_{R,I}$ Automaton

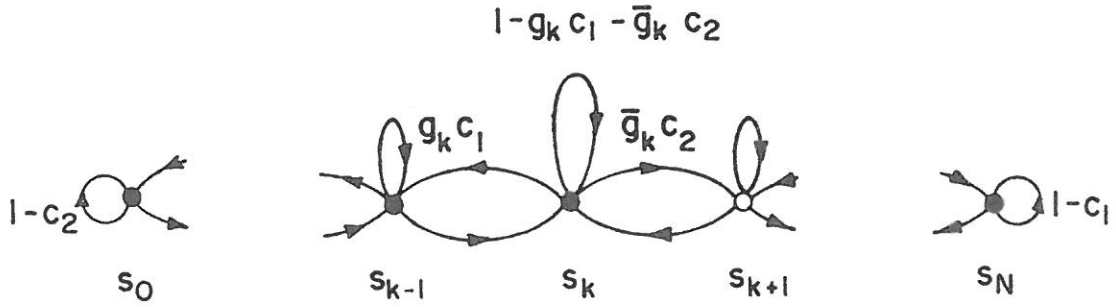


FIG 4: THE DL_{IP} AUTOMATON

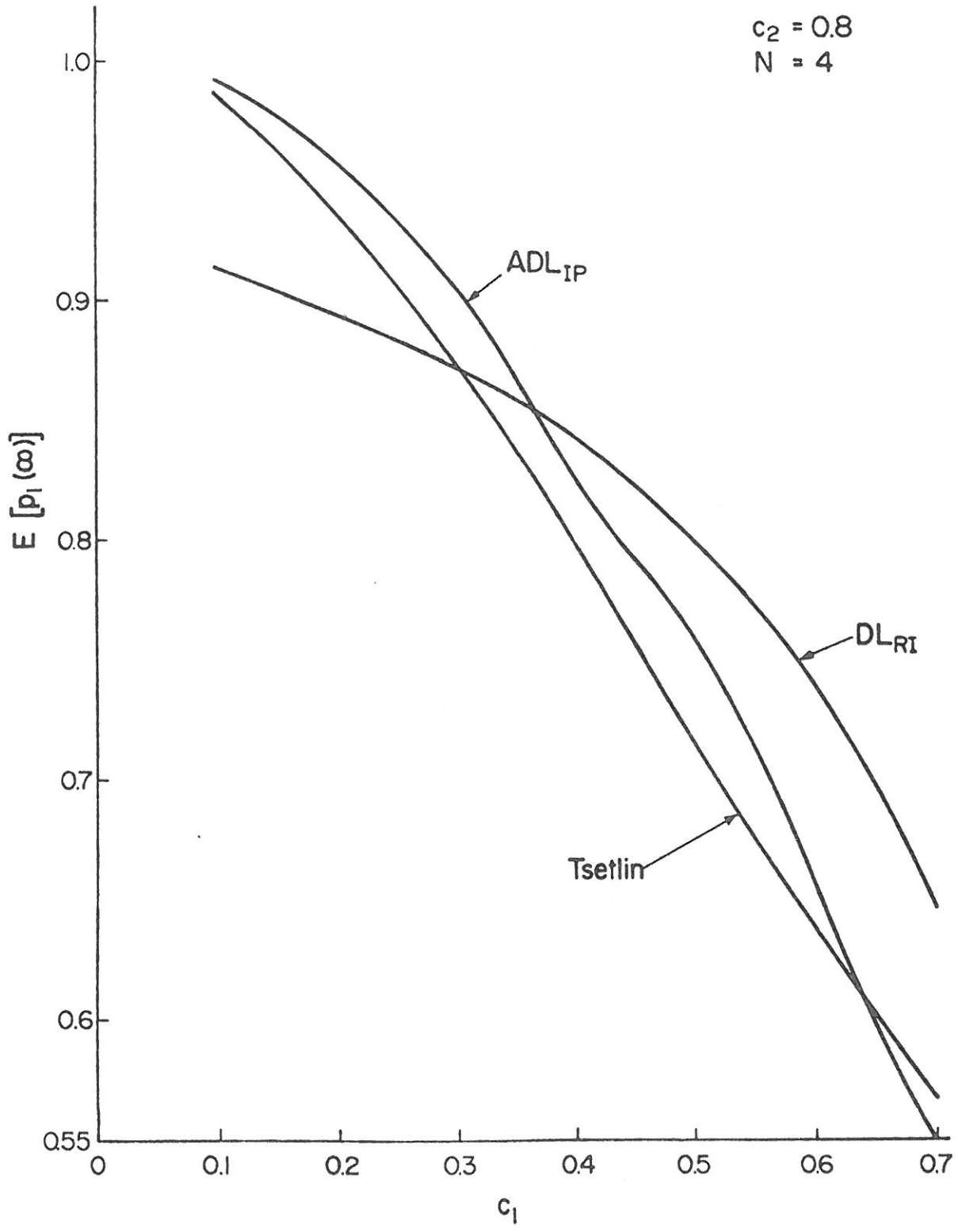


Fig. 5 A relative comparison of the Tsetlin's Automaton, the DL_{RI} scheme and the ADL_{IP} scheme for N = 4.

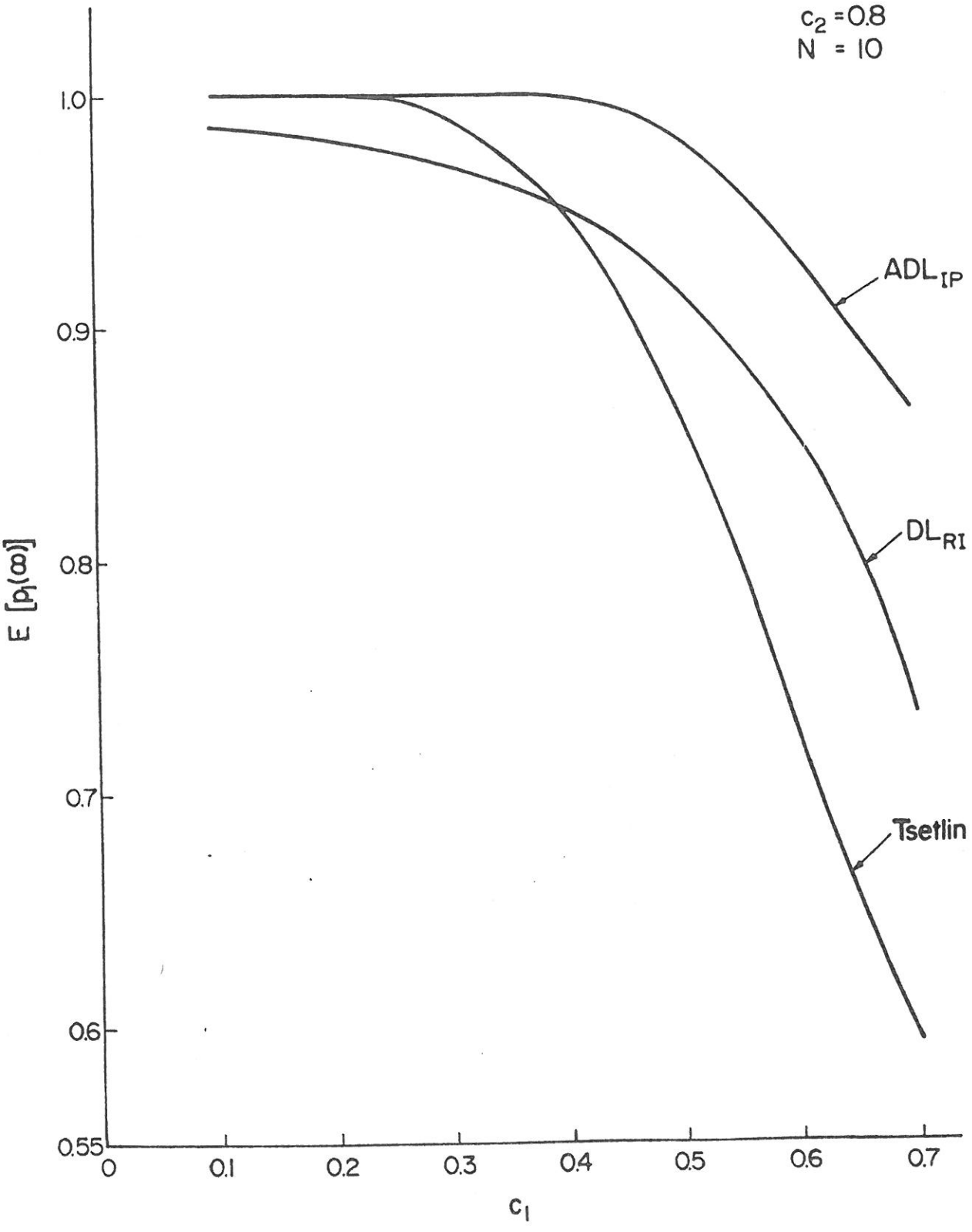


Fig 6 : A relative comparison of the Tsetlin Automaton, the DL_{RI} scheme and the ADL_{IP} scheme for N = 10.

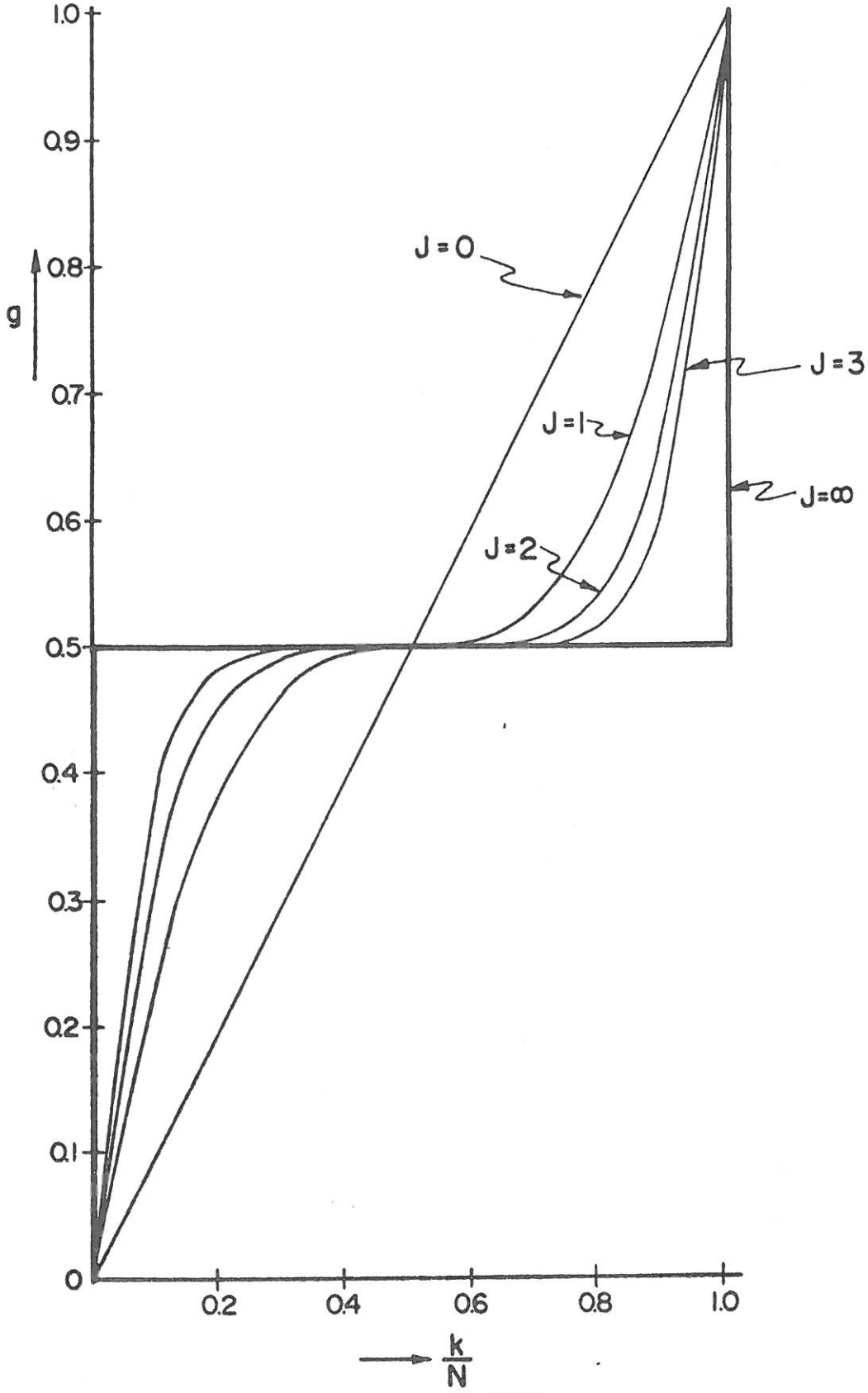


FIG. 7.: A plot of $g = 0.5\left[1+\left(\frac{2k}{N}-1\right)^{2J+1}\right]$ as a function of $\frac{k}{N}$ for various values of J.

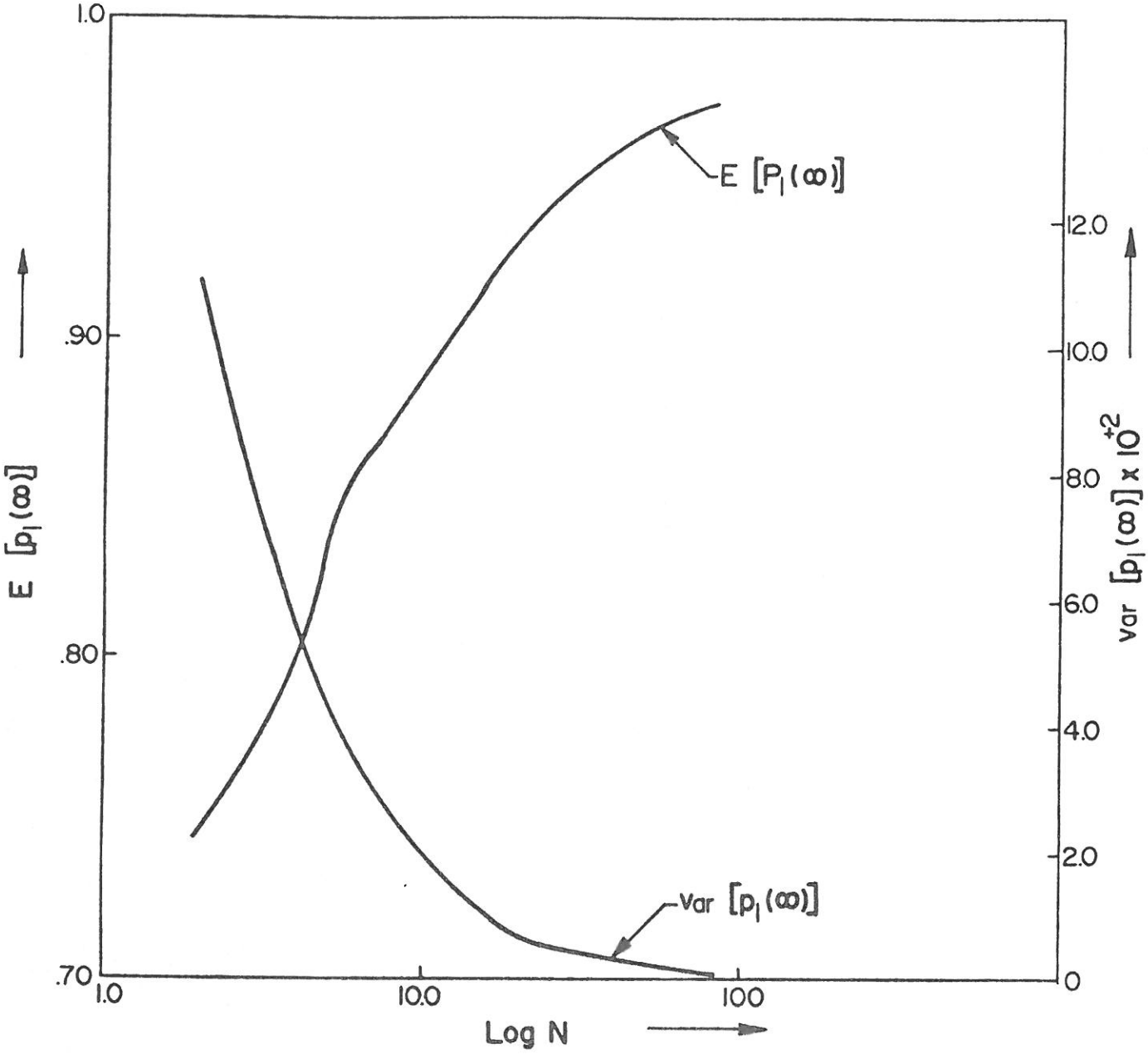


FIG. 8.: Variation of $E[p_1(\infty)]$ and var $[p_1(\infty)]$ with N for the DL_{RP} Automaton. In this case $c_1 = 0.4$ and $c_2 = 0.8$.