ε-OPTIMAL DISCRETIZED LINEAR REWARD-PENALTY LEARNING AUTOMATA

B.J. Oommen* and J.P.R. Christensen**

SCS-TR-112

May 1987

School of Computer Science
Carleton University
Ottawa, Ontario
CANADA K1S 5B6

*School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6
**Københavns Universitets Matematiske Institut, Universitetsparken, 2100 København, DENMARK

This work was partially supported by the Natural Sciences and Engineering Council of Canada. A preliminary version of this paper will be presented as an invited paper at the 1987 IEEE International Conference on Systems, Man and Cybernetics, Alexandria, Virginia, October 1987.

ε–OPTIMAL DISCRETIZED LINEAR REWARD-PENALTY LEARNING AUTOMATA+

B. J. Oommen* and J. P. R. Christensen**

ABSTRACT

In this paper we consider Variable Structure Stochastic Automata (VSSA) which interact with an environment and which dynamically learns the optimal action which the automaton offers. Like all VSSA the automata are fully defined by a set of action probability updating rules [4,9,22]. However, to minimize the requirements on the random number generator used to implement the VSSA, and to increase the speed of convergence of the automaton, we consider the case in which the probability updating functions can assume only a **finite** number of values. These values discretize the probability space [0,1] and hence they are called Discretized Learning Automata. The discretized automata are linear because the sub-intervals of [0,1] are of equal length. We shall prove the following results: (i) Two-Action Discretized Linear Reward-Penalty Automata are ergodic and ε-optimal in all enviroments whose minimum penalty probability is less than 0.5. (ii) There exist Discretized Two-Action Linear Reward-Penalty Automata which are **ergodic** and ε-optimal in **all** random environments. (iii) Discretized Two-Action Linear Reward-Penalty Automata with artificially created absorbing barriers are ε-optimal in **all** random environments.

Apart from the above theoretical results simulation results will be presented which indicate the properties of automata discussed. The rate of convergence of all these automata and some open problems are also presented.

+ Partially supported by the Natural Sciences and Engineering Research Council of Canada. A preliminary version of this paper will be presented as an invited paper at the 1987 IEEE International Conference on Systems, Man and Cybernetics, Alexandria, Virginia, October, 1987.

* School of Computer Science, Carleton University, Ottawa, ONT : K1S 5B6, CANADA.

** Københavns Universitets Matematiske Institut, Universitetsparken, 2100 København, DENMARK.

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 Narenda and Thathachar [9] and an excellent book by Lakshmivarahan [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]. Applications not found in [3] include the solution of stochastic geometric problems using learning automata [15] and the partitioning of objects using various types of automata [16,17].

Broadly speaking, learning automata can be classified into two categories : Fixed Structure Stochastic Automata (FSSA), and Variable Structure Stochastic Automata (VSSA). A Fixed Structure Stochastic Automaton (FSSA) is one whose transition and output functions are time invariant. Examples of such automata are the Tsetlin, Krylov and Krinsky automata [19,20]. 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 [8,9,22].

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 range [0,1]. Hence the RNG 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.

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 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.

Learning automata can also be broadly classified in terms of their Markovian representations. Generally speaking, learning automata are either ergodic [10,13,14-17,19] 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 non-stationary 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].

In this paper we shall be presenting some new results on discretized automata. Historically, various experimental results involving discretized Reward-Inaction automata were first reported by Thathachar and Oommen [18]. 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. Later, in [14] Oommen developed and presented results involving linear and non-linear discretized automata. Among the results proved in [14] were the following 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 (DP_{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 **all** random environments. The latter is the only scheme known to us which is of a linear 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, the maximum advantage that can be obtained by nonlinearizing the automaton was also derived.

In this paper we shall extend the results of [14] and consider various other families of linear discretized automata which are of a Reward-Penalty flavour. We shall prove the following results :

(i) Two-Action Discretized Linear Reward-Penalty (DL_{RP}) automata are ergodic and ε-optimal in all random environments whenever $c_{min} < 0.5$.

(ii) There exist Two-Action Discretized Linear Reward-Penalty Automata which are **ergodic** and ε-optimal in **all** random environments. We shall refer to this machine as the Modified Discretized Linear Reward-Penalty (MDL_{RP}) automata.

(iii) Discretized Two-Action Linear Reward-Penalty Automata with artificially created absorbing barriers are ε-optimal in all random environments. These automata shall be called the Absorbing Discretized Linear Reward-Penalty (ADL_{RP}) automata.

The above automata are the **only** schemes known to us which are of a linear nature and yet ε-optimal even though the probability re-enforcing rules are of a reward-penalty flavour.

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 Section IV of [14], 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 above schemes is a major contribution. Although **continuous** linear symmetric Reward-Penalty schemes are at their best expedient (and definitely **not** absolutely expedient [3]) reward-penalty schemes are not entirely rejectable. In this paper, we have shown that by discretizing the probability space and rendering the boundary values **absorbing** the resulting symmetric automaton is indeed ε-optimal. Alternatively, by making a stochastic modification to the transition function, the automata can be made **ergodic** and ε-optimal in all random environments.

The question naturally arises : Are there situations in which a reward-penalty scheme is to be preferred ? Simulation results indicate that the ADL_{RP} scheme is **extremely** accurate and fast in its convergence. Further, in a case when the penalty probabilities are near 0.5 (i.e. the reward probabilities are almost the same as the penalty probabilities), the automaton utilizes all the responses of the environment and ignores none of the environment responses as a reward-inaction automaton does.

For the rest of this section we shall present some fundamentals and the notation we shall be using. We shall subsequently present the various theoretical results we have obtained concerning the three automata discussed above. We shall then present the simulation results and compare the corresponding automata with known existing learning machines. We shall conclude the paper with the simulation results of the DL_{RP} automaton iterating with a non-stationary environment.

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 $\{ a_i \mid i = 1 \text{ to } R \}$. The selection is done on the basis of a probability distribution $\mathbf{p}(n)$, an R x 1 vector where, $\mathbf{p}(n) = [p_1(n), p_2(n), \ldots, p_R(n)]^T$ with,

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

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

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,

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

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

The reward probability is defined as $1 - c_i$ for $1 \le i \le R$.

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 L is the action which obeys,

$$c_L = \min_i (c_i) \tag{3}$$

then $p_L(n) = 1$, $p_i(n) = 0$ for $i \neq L$ achieves this result. Updating schemes for **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. The analogous results for R > 2 are yet open but conjectured to be true.

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-PENALTY (DL_{RP}) AUTOMATON

The Discretized Linear Reward-Penalty (DL_{RP}) 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 i / N, and this represents the probability of the automaton choosing action a_1. Note that in this state the automaton chooses action a_2 with probability (1- 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. The transition map of the DL_{RP} automaton is specified by (4) below for $s(n) = s_k$, $1 \leq k \leq N-1$.

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

Observe that (4) is valid only for the interior states. For the end states :

$$
\begin{aligned}
s(n+1) &= s(n) && \text{if } s(n) = s_0 \text{ or } s_N \text{ and } b(n) = 0 \\
&= s_1 && \text{if } s(n) = s_0 \text{ and } b(n) = 1 \\
&= s_{N-1} && \text{if } s(n) = s_N \text{ and } b(n) = 1.
\end{aligned}
$$

Figure II shows the transition map of the automaton schematically.

Observe that if 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 :

$$
\begin{aligned}
p_1(n+1) &= p_1(n) + 1/N && \text{if } a_1 \text{ is chosen and } b(n) = 0 \\
&&& \text{or } a_2 \text{ is chosen and } b(n) = 1 \\
&= p_1(n) - 1/N && \text{if } a_1 \text{ is chosen and } b(n) = 1 \\
&&& \text{or } a_2 \text{ is chosen and } b(n) = 0.
\end{aligned} \tag{5}
$$

At the end states the following equality holds :

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

The way by which the action probabilities are updated warrants the name of the automaton.

If $c_1 < c_2$, the automaton has no absorbing barriers except in the degenerate cases when $c_1 = 0$ or $c_2 = 1$. 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]. More specifically, $p_1(n)$ behaves as a homogeneous Markov chain defined by a stochastic matrix M whose arbitrary element $M_{i,j}$ is defined as :

$$
M_{i,j} = \Pr[\, s(n) = s_j \mid s(n-1) = s_i \,], \quad \text{where,}
$$

$$
\begin{aligned}
M_{i,i-1} &= g_i c_1 + g'_i (1 - c_2) && \text{for } 1 \le i \le N, \\
M_{i,i+1} &= g'_i c_2 + g_i (1 - c_1) && \text{for } 0 \le i \le N-1, \\
M_{i,i} &= 0 && \text{for } 1 \le i \le N-1 \qquad (6)
\end{aligned}
$$

where $g_i = i / N$ and $g'_i = 1 - i / N$. All the other elements of M are zero. Furthermore, the boundary conditions for the Markov chain are specified by :

$$M_{0,0} = (1 - c_2) \quad \text{and} \quad M_{N,N} = (1 - c_1). \tag{7}$$

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 $\pi(n)$ be the state probability vector, where, for all n,

$$\pi(n) = [\pi_0(n), \pi_1(n), \ldots, \pi_N(n)]^T, \qquad \pi_i(n) = \Pr[s(n) = s_i], \text{ and,} \tag{8}$$

$$\sum_{i=0}^{N} \pi_i(n) = 1.$$

Then the limiting value of π is given by the vector which satisfies,

$$M^T \pi = \pi \tag{9}$$

Using (9) we now derive the asymptotic properties of the DL_{RP} automaton.

Theorem I.

Let $\Delta = (c_1 + c_2 - 1)$. Then π_i, the ith component of the asymptotic probability vector obeys the following difference equation for $1 \le i \le N$.

$$\pi_i = \frac{c_2 - \Delta\left(\frac{i-1}{N}\right)}{(1-c_2) + \Delta\frac{i}{N}} \pi_{i-1} \qquad i = 1, 2, \ldots, N.$$

Proof :

By definition, the limiting equilibrium probability vector π satisfies

$$M^T \pi = \pi,$$

where, π is defined by (8) above. To render the computations easy we introduce the following polynomials P(Z) and Q(Z), where,

$$P(Z) = \sum_{i=0}^{N} \pi_i Z^i, \qquad \text{and}$$

$$Q(Z) = \frac{1}{N} \cdot Z \cdot P'(Z) = \sum_{i=0}^{N} \frac{i}{N} \cdot \pi_i \cdot Z^i .$$

Using the notation that $\Delta = c_1 + c_2 - 1$, the equation $\pi = M^T \pi$ can be easily seen to be equivalent to (10) below :

$$\left(\frac{\Delta}{Z} - \Delta Z\right) Q(Z) + \left(\frac{1 - c_2}{Z} + c_2 Z\right) P(Z)$$

$$- (1 - c_2) \frac{\pi_0}{Z} + \pi_0 (1 - c_2) - c_2 \pi_N Z^{N+1} + (1-c_1)\pi_N Z^N + (c_1+c_2-1)\pi_N Z^{N+1} = P(Z) \qquad (10)$$

Moving P(Z) to the left hand side, multiplying by Z and **dividing** by 1 - Z yields :

$$\Delta(1 + Z) Q(Z) + ((1 - c_2) - c_2 Z) P(Z) = (1 - c_2) \pi_0 - (1 - c_1) \pi_N Z^{N+1} \qquad (11)$$

By comparing coefficients this gives

$$\left(\Delta \frac{i}{N} + (1 - c_2)\right) \pi_i + \left(\Delta \frac{i-1}{N} - c_2\right) \pi_{i-1} = 0 \qquad i = 1, 2, \ldots, N \qquad (12)$$

Moving terms with π_{i-1} to the right side and dividing by the coefficient of π_i yields:

$$\pi_i = \frac{c_2 - \Delta\left(\frac{i-1}{N}\right)}{(1-c_2) + \Delta \frac{i}{N}} \pi_{i-1} \qquad i = 1, 2, \ldots, N. \qquad (13)$$

and the theorem is proved. •••

We shall now prove the ε-optimal properties of the DL_{RP} scheme.

Theorem II.

The DL_{RP} automaton is ε-optimal whenever the minimum penalty probability is less than 0.5.

Proof :

With no loss of generality let a_1 be the optimal action (i.e., let $c_1 < c_2$). It is remains to be proved that $E[p_1(\infty)]$ tends to unity as $N \to \infty$ if and only if $c_1 < 0.5$, where,

$$E[p_1(\infty)] = \sum_{i=0}^{N} \left(\frac{i}{N}\right) \pi_i$$

We consider three mutually exclusive and exhaustive cases.

Case I : $c_2 > 0.5 > c_1$.

From (13) we can see that if $c_2 > 0.5 > c_1$, then,

$$\pi_i \geq q\, \pi_{i-1}$$

where $q > 1$ for all i.

This easily implies that as $N \to \infty$ the major part of the probability measure on π is contained in an arbitrarily small neighbourhood of unity. Thus,

$$\lim_{N\to\infty} E[p_1(\infty)] = \lim_{N\to\infty} \sum_{i=0}^{N} \frac{i}{N} \pi_i \quad \to \quad 1.$$

Case II : $0.5 \geq c_2 > c_1$.

$$\text{Let } i_0 = N\left[\frac{1}{2\ N} + \frac{1-2c_2}{2(1-(c_1+c_2))}\right].$$

Note that i_0 need not be an integer. For the sake of notation, let the ratio of π_i to π_{i-1} in (13) be q_i. Then, a simple algebraic computation shows that :

$$q_i = \frac{c_2 - \Delta(\frac{i-1}{N})}{(1-c_2) + \Delta(\frac{i}{N})} \qquad < 1 \qquad \text{for } 1 \leq i < i_0$$

$$> 1 \qquad \text{for } i_0 < i \leq N.$$

It is important to observe that for large N, $i_0 < qN$, where q is strictly less than 0.5.

The ε-optimality of the scheme when $c_2 = 0.5$ is disposed of by remarking that q_i increases to $(1-c_1+\Delta/N)\,/\,c_1$ and is strictly greater than unity for all i.

Consider now the case when $c_2 < 0.5$. In this case, $q_i < 1$ and increases for $1 \leq i < i_0$, and continues to increase beyond i_0. We compare π_i for $0 \leq i < i_0$ and $i_0 < i < 2i_0 + 2$ to π_i for i in the interval $2i_0 + 2 < i \leq N$. In the latter interval,

$$q_i > \frac{1-c_2}{c_2} > 1.$$

Let i_1 be the first integer in the interval $(2i_0 + 2, N]$. Then the probability measure in the first two intervals sum to a quantity **less** than $2qN\pi_1$, where q is chosen strictly less than 0.5 such that $i < qN$ for sufficiently large N and q independent of N. Similarly, the probability measure in the last interval sums to a quantity **more** than S', where,

$$S' = \frac{1 - \left(\frac{1 - c_2}{c_2}\right)^{(1-2q)N}}{1 - \frac{1 - c_2}{c_2}} \cdot \pi_i \tag{14}$$

This shows that for $N \to \infty$ most of the probability mass sits in the last interval, and an argument as in case (i) finishes the proof. Between Cases I and II we see that the scheme is ε-optimal whenever $c_1 < 0.5$.

Case III : $c_2 > c_1 \geq 0.5$.

Let α be defined as $\alpha = (2c_2 - 1)/2(c_2 + c_1 - 1)$. Of course, α is strictly greater than 0.5 in the case we are considering. Let $d > 0$ be an arbitrarily small positive number. For $i = 0,\ldots, N$, let i_1 be the first of the numbers i/N which belong to the interval from α-d to α, and let $\pi(i_1)$ be the corresponding associated probability measure. Since the probabilities are increasing in the interval from 0 to i_1 the probability of the whole interval from 0 to α-d is bounded **above** by $i_1 . \pi(i_1)$. We shall show that the probability of the interval from α-d to α-d/2 is bounded **below** by $\pi(i_1)$ times the sum of a quotient series where the quotient is bounded **below** by $d(c_1+c_2-1)+1$ independent of N (if d is sufficiently small and N is large enough). But the number of terms in that quotient series is asymptotic to (1/2d) N since each of the numbers in the intersection of the progression i/N with the interval from α-d to α - d/2 contributes one term. Hence, as N tends to infinity, most of the probability mass in the interval from 0 to α sits in the interval from α-d to α. A very similar argument gives that most of the probability mass in the interval from α to 1 sits in the interval from α to α+d, where d is arbitrarily small. This concludes the proof.

Hence the automaton is not ε-optimal whenever the minimum penalty probability is greater than 0.5. Interestingly enough the value of $E[p_1(\infty)] = 1$ when $c_1 = 0.5$.

Hence the theorem ! ●●●

Remarks.

1. The question of whether the DL_{RP} automaton was ε-optimal was left open in [14], but Oommen conjectured that the machine was ε-optimal in **all** environments. An

anonymous reviewer of [14] had suggested that the conjecture was too powerful, and indeed this is the case (see the footnote of page 291 of [14]). The first author of this paper would like to put on record his gratitude to the reviewer of [14] who pointed him to the true property of the DL_{RP} automaton.

2. When Tsetlin first designed the Tsetlin automaton, $L_{2N,2}$ (or linear tactic), his automaton was the first (deterministic or stochastic) automaton that could be proven to possess learning properties. The automaton was shown to be ε-optimal in environments whenever the minimum penalty probability is less than 0.5. It is not inappropriate to mention that the DL_{RP} automaton is **not** a generalized version of the linear tactic, but is distinct in both design and operation for the following reasons :

(a) Whereas the $L_{2N,2}$ automaton is a FSSA, the DL_{RP} scheme is a VSSA.

(b) In the case of the $L_{2N,2}$ automaton, the action probability vector is a **deterministic** vector. In the case of the DL_{RP} scheme ,**p**(n) is a random vector. Thus, whenever $c_1 < 0.5$, whereas in the former case the **probability** $p_1(\infty) \to 1$ as $N \to \infty$, in the latter case the **expected** probability $E\ [p_1(\infty)] \to 1$ as $N \to \infty$. Thus all the advantages of VSSA over FSSA (such as that of possessing the capability of choosing different actions at almost all consecutive time instances) are found in the DL_{RP} scheme. Additionally, the expected penalty tends to the value of the minimum penalty probability whenever the latter quantity is less than 0.5.

(c) When interacting with non-stationary environments, we shall show that the DL_{RP} scheme is superior to the $L_{2N,2}$ automaton.

We shall now present a modification of the DL_{RP} automaton which is ε-optimal in **all** random environments.

III. THE MODIFIED DISCRETIZED LINEAR REWARD-PENALTY (MDL_{RP}) AUTOMATON

The Modified Discretized Linear Reward-Penalty (MDL_{RP}) automaton has (N+1) states where N is an **even** integer, which as in the case of the DL_{RP} automaton is the set $S = \{s_0, s_1, \ldots, s_N\}$. Associated with the state s_i is the probability i/N, and this represents the probability of the automaton choosing action a_i. As before, note that in this state the automaton chooses the action a_2 with probability (1 - i/N).

As in Section II, the learning is achieved by making **discrete** changes in the action probabilities, and is done by defining the transition map as a function from S X B to S. The transition map of the MDL_{RP} automaton is specified **stochastically** below for $s(n) = s_k$, $1 \le k \le N-1$.

$$
\begin{aligned}
s(n+1) &= s_{k+1} && \text{w.p. } 1.0 && \text{if} && a(n) = a_1 \text{ and } b(n) = 0 \\
&= s_{k+1} && \text{w.p. } 0.5 && \text{if} && a(n) = a_1 \text{ and } b(n) = 1 \\
&= s_{k+1} && \text{w.p. } 0.5 && \text{if} && a(n) = a_2 \text{ and } b(n) = 1 \\
&= s_{k-1} && \text{w.p. } 1.0 && \text{if} && a(n) = a_2 \text{ and } b(n) = 0 \\
&= s_{k-1} && \text{w.p. } 0.5 && \text{if} && a(n) = a_2 \text{ and } b(n) = 1 \\
&= s_{k-1} && \text{w.p. } 0.5 && \text{if} && a(n) = a_1 \text{ and } b(n) = 1
\end{aligned}
\tag{15}
$$

At the boundary states the MDL_{RP} automaton obeys :

$$
\begin{aligned}
s(n+1) &= s(n) && \text{w.p. } 1.0 && \text{if} && s(n) = s_0 \text{ or } s_N \text{ and } b(n) = 0 \\
&= s(n) && \text{w.p. } 0.5 && \text{if} && s(n) = s_0 \text{ or } s_N \text{ and } b(n) = 1 \\
&= s_1 && \text{w.p. } 0.5 && \text{if} && s(n) = s_0 \text{ and } b(n) = 1 \\
&= s_{N-1} && \text{w.p. } 0.5 && \text{if} && s(n) = s_N \text{ and } b(n) = 1.
\end{aligned}
\tag{16}
$$

Observe that if 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 :

$$
\begin{aligned}
p_1(n+1) &= p_1(n) + 1/N && \text{w.p. } 1.0 && \text{if} && a(n) = a_1 \text{ and } b(n) = 0 \\
&= p_1(n) + 1/N && \text{w.p. } 0.5 && \text{if} && a(n) = a_1 \text{ and } b(n) = 1 \\
&= p_1(n) + 1/N && \text{w.p. } 0.5 && \text{if} && a(n) = a_2 \text{ and } b(n) = 1 \\
&= p_1(n) - 1/N && \text{w.p. } 1.0 && \text{if} && a(n) = a_2 \text{ and } b(n) = 0 \\
&= p_1(n) - 1/N && \text{w.p. } 0.5 && \text{if} && a(n) = a_2 \text{ and } b(n) = 1 \\
&= p_1(n) - 1/N && \text{w.p. } 0.5 && \text{if} && a(n) = a_1 \text{ and } b(n) = 1.
\end{aligned}
\tag{17}
$$

At the boundary states the probability changes as below :

$$
\begin{aligned}
p_1(n+1) &= p_1(n) && \text{w.p. } 1.0 && \text{if} && s(n) = s_0 \text{ or } s_N \text{ and } b(n) = 0 \\
&= p_1(n) && \text{w.p. } 0.5 && \text{if} && s(n) = s_0 \text{ or } s_N \text{ and } b(n) = 1 \\
&= 1/N && \text{w.p. } 0.5 && \text{if} && s(n) = s_0 \text{ and } b(n) = 1 \\
&= (N-1)/N && \text{w.p. } 0.5 && \text{if} && s(n) = s_N \text{ and } b(n) = 1.
\end{aligned}
\tag{18}
$$

We shall now prove the ε-optimal properties of the MDL_{RP} scheme.

Theorem III.

The MDL_{RP} automaton defined by (15) and (16) is ε-optimal in **all** random environments.

Proof :

If $c_1 < c_2$, the automaton has no absorbing barriers except in the degenerate cases when $c_1 = 0$ or $c_2 = 1$. This implies that without loss of generality the Markov chain is ergodic. $p_1(n)$ behaves as an ergodic homogeneous Markov chain defined by a stochastic matrix Q whose arbitrary element $Q_{i,j}$ is defined as

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

$$\begin{aligned}
Q_{i,i-1} &= 0.5g_ic_1 + g'_i(1 - c_2) + 0.5\, g'_ic_2 && \text{for } 1 \le i \le N, \\
Q_{i,i+1} &= 0.5g_ic_1 + g_i\,(1 - c_1) + 0.5g'_ic_2 && \text{for } 0 \le i \le N-1, \\
Q_{i,i} &= 0 && \text{for } 1 \le i \le N-1 \quad (19)
\end{aligned}$$

where $g_i = i / N$ and $g'_i = 1 - i / N$. All the other elements of Q are zero.

The boundary conditions for the Markov chain are specified by :

$$Q_{0,0} = 0.5c_2 + (1-c_2) \qquad \text{and} \qquad Q_{N,N} = 0.5c_1 + (1-c_1). \qquad (20)$$

Let $e_1 = 0.5c_1$ and $e_2 = 0.5c_2$. Then, (19) and (20) become (21) and (22) respectively.

$$\begin{aligned}
Q_{i,i-1} &= g_ie_1 + g'_i(1 - e_2) && \text{for } 1 \le i \le N, \\
Q_{i,i+1} &= g'_ie_2 + g_i\,(1 - e_1) && \text{for } 0 \le i \le N-1, \\
Q_{i,i} &= 0 && \text{for } 1 \le i \le N-1 \quad (21)
\end{aligned}$$

$$Q_{0,0} = 1 - e_2 \qquad \text{and} \qquad Q_{N,N} = 1 - e_1 \qquad (22)$$

Comparing (21) and (22) with (6) and (7) we observe that :

(i) There is a non-zero entry in Q if and only if there is one in the corresponding position in M.

(ii) Every c_i in M is replaced by e_i (i.e. $0.5c_i$) in Q. Similarly $(1-c_i)$ in M is replaced by $(1 - e_i)$ in Q.

Due to the above observations, the ergodic Markov chain represented by Q can be solved trivially, by merely substituting in the solution for (6) and (7) e_i and $1-e_i$ instead of c_i and $1-c_i$ respectively. This leads us to the interesting conclusion that the

MDL_{RP} automaton interacting with an environment with penalty probabilities (c_1, c_2) behaves exactly as a DL_{RP} automaton would if it interacted with an environment with penalty probabilities ($c_1/2$, $c_2/2$). Since c_1 and c_2 are probabilities the ε-optimality of the MDL_{RP} automaton in **all** environments follows from the ε-optimality properties of the DL_{RP} automaton proved in Theorem II. Hence the result ! •••

Remarks .

1. The DL_{RP} automaton is the only known **ergodic symmetric linear** Reward-Penalty VSSA which is ε-optimal in **any** random environment. In the continuous case, it is easy to see that no such scheme can exist - since the symmetric L_{RP} scheme is at its best expedient [3]. By discretizing the probability space and by rendering the probability changes discrete the automaton can be made ε-optimal in some environments. This, in our opinion, in itself, is a significant discovery not only in the field of adaptive learning but also in the area of the psychological modelling of biological systems.

2. The MDL_{RP} automaton is the only known **ergodic linear** ε-optimal reward-penalty VSSA which does not require the penalty response to be **arbitrarily** smaller than the response to a reward. This is notably distinct from the set of ergodic ε-optimal schemes described in [3].

3. The MDL_{RP} scheme can be viewed as a filter in conjunction with the DL_{RP} scheme described in Section II. The filter transforms the responses of the environments as follows : Whenever $b(n) = 0$ the filter emits the response $b'(n)$ identically equal to 0. However, whenever $b(n) = 1$, the filter emits the response $b'(n)$ to be 1 with a probability of 0.5, and emits the response $b'(n)$ to be equal to 0 with a probability of 0.5. This conceptual view of the MDL_{RP} scheme is shown in Figure III. Notice that although the environment may have the penalty probabilities (c_1, c_2), the DL_{RP} automaton effectively interacts with a "pseudo-environment" with penalty probabilities ($c_1/2$, $c_2/2$). We call such a filter an " Environment Transforming Filter ". We are currently investigating the existence of various other such filters and studying their application to list organizing strategies.

We shall now proceed to present a symmetric linear reward-penalty scheme which is **ε-optimal** in all random environments.

IV. THE ABSORBING DISCRETIZED LINEAR REWARD-PENALTY (ADL_{RP}) SCHEME

The Absorbing Discretized Linear Reward-Penalty (ADL_{RP}) automaton is obtained by defining the states s_0 and s_N of the DL_{RP} 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_{RP} automaton, and,

(ii) G is the state transition map specified by (23) below for $s(n) = s_k, 1 \leq k \leq N-1$.

$$
\begin{aligned}
s(n+1) &= s_{k+1} && \text{if } a(n) = a_1 \text{ and } b(n) = 0, \\
&&& \text{or } a(n) = a_2 \text{ and } b(n) = 1 \\
&= s_{k-1} && \text{if } a(n) = a_1 \text{ and } b(n) = 1, \\
&&& \text{or } a(n) = a_2 \text{ and } b(n) = 0.
\end{aligned} \tag{23}
$$

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.

Notice that, as in the case of the DL_{RP} scheme, if the machine is in state s_i, it will choose action a_1 with probability i/N. Thus in this state it chooses a_2 with probability (1-i/N). The above design warrants the name of the automaton.

As in the other cases recorded earlier, observe that if 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 :

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

At the end states the following equality holds for all n.

$$p_1(n+1) = p_1(n) \qquad \text{if } p_1(n) \in \{0, 1\}.$$

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 R be the stochastic matrix defining the chain, whose arbitrary element $R_{i,j}$ is :

$$R_{i,j} = Pr[\, s(n) = s_j \mid s(n-1) = s_i \,] \text{ , and,}$$

$$\begin{aligned} R_{i,i-1} &= g_i c_1 + g'_i (1 - c_2) && \text{for } 1 \le i \le N, \\ R_{i,i+1} &= g'_i c_2 + g_i (1 - c_1) && \text{for } 0 \le i \le N-1, \\ R_{i,i} &= 0 && \text{for } 1 \le i \le N-1 \end{aligned} \qquad (24)$$

where $g_i = i / N$ and $g'_i = 1 - i / N$. All the other elements of R are zero, excepting $R_{0,0}$ and $R_{N,N}$. Since the chain is absorbing, $R_{0,0} = R_{N,N} = 1$.

We can now prove the asymptotic properties of the ADL_{RP} scheme.

Theorem IV.

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

Proof :

With no loss of generality let let a_1 be the optimal action (i.e., let $c_1 < c_2$). Let H(i) be the first passage probability of being absorbed in state s_N given that the chain started in state s_i. Clearly,

$$H(0) = 0, \quad \text{and,} \quad H(N) = 1.$$

Additionally, $0 \le H(i) \le 1$. Assuming that we start at state N/2, we aim to prove that H(N/2) tends to unity as $N \to \infty$.

Let x_i be the difference between H(i) and H(i - 1) for $1 \le i \le N$. Then, since H is invariant under the chain, we get :

$$\left[\frac{i}{N} c_1 + \left(1 - \frac{i}{N}\right)(1 - c_2)\right] x_i = \left[\frac{i}{N} (1 - c_1) + \left(1 - \frac{i}{N}\right) c_2\right] x_{i+1} \qquad 1 \le i \le N-1 \qquad (25)$$

Thus x_i is a probability distribution on i and H(i) is the cumulative probability. Rewriting (25) yields,

$$x_{i+1} = \frac{\frac{i}{N} c_1 + \left(1 - \frac{i}{N}\right)(1 - c_2)}{\frac{i}{N}(1 - c_1) + \left(1 - \frac{i}{N}\right) c_2}\, x_i \qquad 1 \le i \le N-1 \qquad (26)$$

The proof now follows, very closely, the proof of Theorem II, the main difference being that the quotients are (very close to) the reciprocals of the quotients obtained in

Theorem II. Hence we shall show that most of the probability mass of the vector $\mathbf{x} = [0, x_1, x_2, \ldots, x_N]^T$ lies in the "head" (i.e. in the leftmost components of $\mathbf{x}$). We shall consider three distinct cases as in Theorem II.

Case I. $c_2 > 0.5 > c_1$.

In this case we see that

$$x_{i+1} < q\, x_i \qquad\qquad 1 \le i \le N - 1,$$

$$\text{where } q = \max\left[\frac{1 - c_2}{c_2}, \frac{c_1}{1 - c_1}\right]. \tag{27}$$

Since $q < 1$, as $N \to \infty$, most of the probability mass of the vector $\mathbf{x}$ sits at the head and hence H(i) **tends to unity** when i/N is bounded away from zero. In particular, of course, H(N/2) tends to unity.

Case II. $0.5 \ge c_2 > c_1$.

In this case we define $\alpha = (2c_2 - 1)/\ 2(c_1 + c_2 - 1)$. Then $0 \le \alpha < 0.5$, with the inequality being strict at the upper bound. The value of the quotients q_i starts (for i = 0) by being equal to $(1 - c_2)/c_2 > 1$, then descends for $i/N < \alpha$ to the value of unity. It further descends to $c_1/(1 - c_1) < 1$ for i/N in the interval from α to 1. The situation is thus very similar to Case III of Theorem II and the proof is almost identical. Thus in this case, most of the probability mass of the vector $\mathbf{x}$ sits at i, where i/N is in the neighbourhood of α. Hence, as $N \to \infty$, if $i/N \ge \alpha + \varepsilon_0$ (with $\varepsilon_0 > 0$) H(i) tends to unity. In particular again, of course, H(N/2) tends to unity as $N \to \infty$.

Case III. $c_2 > c_1 > 0.5$.

In this case a simple algebraic manipulation shows that x_i decreases for $i/N < \alpha$, where α, as before, is defined by $(2c_2 - 1)/\ 2(c_1 + c_2 - 1)$, and satisfies $\alpha > 0.5$. Let $i_0=\alpha$. In this case, the quotient starts by being $(1 - c_2)/c_2 > 1$, and ascends to 1 in the interval from 0 to α, then further ascends to $c_1/(1 - c_1) > 1$ in the interval from α to 1. We prove that most of the probability sits in the leftmost part of the unit interval. The argument is very similar to Case II of Theorem II except that the unit interval is divided in the three subintervals $[0, 2i_0-1]$; $[2i_0-1, i_0]$ and finally $[i_0,1]$. Most of the probability sits in the first interval and the proof is essentially the same, the difference being that i_0 is precisely equal to α and the situation has been mirrored into the middle point 1/2. Consequently it can be seen that as $N \to \infty$ the probability mass in the first interval far exceeds the probability mass in the other two intervals. Thus, most of the mass sits in the left most portion of **x**, and thus $H(i) \rightarrow 1$ as $N \rightarrow \infty$ if $i/N \geq \varepsilon_0 > 0$. In particular, of course, again $H(N/2) \rightarrow 1$ as $N \rightarrow \infty$ and the theorem is proved. ●●●

The ADL_{RP} scheme is the only known **symmetric linear** reward-penalty scheme which is ε-optimal in **all** random environments. We conjecture that there is none other. Indeed, it is **far** superior to its corresponding continuous conterpart.

V. EXPERIMENTAL RESULTS

To evaluate the performance of the DL_{RP} and ADL_{RP} automata, the latter were 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 automata 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 capability of the DL_{RP} scheme as a function of the number of states which it possessed has been tabulated in Table I.

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

Table I : Variation of $\hat{E}[p_1(\infty)]$ 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 $\hat{E}[p_1(\infty)]$ and $Var[p_1(\infty)]$ with N is shown in Figure IV for the case when $c_1 = 0.4$ and $c_2 = 0.8$.

In the case of the ADL_{RP} scheme, simulations were performed with environments just as described above. However, to render the experimental results meaningful, the learning properties of the ADL_{RP} automaton was also compared with two other finite state learning machines - the 2N-state Tsetlin automaton, the corresponding (N+1) State Discretized Linear Reward-Inaction (DL_{RI}) automaton, and the corresponding (N+1) State Absorbing Discretized Linear Inaction-Penalty (ADL_{IP}) automaton for various values of N. Figure V shows the variation of $\hat{E}[\ p_1(\infty)\ ]$ with c_1, for the depths of memory of the machines being 10. Observe the superiority of the ADL_{RP} automaton in all environments for N = 10. Such results are typical.

To compare the rate of convergence and the accuracies of various absorbing discretized automata, we present below some of the experimental results obtained involving the DL_{RI}, the ADL_{IP} and the ADL_{RP} automata. Some typical results are tabulated in Table II. From the table we see that the ADL_{RP} scheme is superior on counts of both speed and accuracy. For example, when N=10, c_1=0.6 and c_2 = 0.8, the DL_{RI} scheme converges with an expected accuracy of 85.5%. The mean time to converge for the DL_{RI} scheme was 25.58 iterations. The corresponding figures for the ADL_{IP} scheme were 93% and 499.11 iterations respectively. However, for the ADL_{RP}, the accuracy was 93% and the mean time to converge was **only** 32.45 iterations.

	C_1	DL_{RI} Scheme $\hat{E}[p_1(\infty)]$	M.T.C.	ADL_{IP} Scheme $\hat{E}[p_1(\infty)]$	M.T.C.	ADL_{RP} Scheme $\hat{E}[p_1(\infty)]$	M.T.C.
N = 4	0.2	0.896	4.61	0.960	9.87	0.95	2.965
	0.4	0.843	6.04	0.825	10.65	0.86	3.730
	0.6	0.741	8.52	0.655	10.57	0.72	4.830
N = 10	0.2	0.980	11.46	1.00	70.28	1.00	8.220
	0.4	0.951	16.15	1.00	196.78	0.98	14.88
	0.6	0.855	25.58	0.93	499.11	0.93	32.45

Table II : Typical results demonstrating the properties of the DL_{RI}, ADL_{IP}, and the ADL_{RP} Automata. In all the experiments c_2 = 0.8.

Of all the linear automata which we have worked with, the ADL_{RP} automaton seems to be the most superior based on counts of both speed and accuracy. It is indeed an extremely impressive learning machine.

We shall now consider the learning properties of the DL_{RP} scheme when interacting with a non-stationary environment. Although the details of these results have been presented elsewhere [14], for the sake of completeness we present the conclusions again in all brevity, so that this paper represents a comprehensive study of the state of the field when it concerns Discretized Reward-Penalty automata.

VI. 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 [19,20]. 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 by $P_{E_i}(n)$, then the probability of being in the same environment in the next instant was given by :

$$(1 - \delta)\, P_{E_i}(n) + \delta\, P_{E_j}(n) \qquad i \neq j;\, i, j = 1, 2. \tag{28}$$

When δ is small, (28) states that with almost the same probability 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 $[p_{E1}, p_{E2}]^T$ 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 $1\text{-}c_1$, in Tsetlin work, E_2 was so chosen to have penalty probabilities $1\text{-}c_1$ 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 $(M_0 - M^*)$ indicates a better automaton.

It was shown by Tsetlin that there was, for each environment, an optimal memory for which $(M_0 - M^*)$ 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).

In [14], Oommen indicated that the DL_{RP} scheme (with N=2) 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 $(M_0 - M^*)$. Based on other experimental results presented elsewhere [14] (they are not repeated here for the sake of brevity), the following observations can be made :

(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 requirement is less.

(iii) From Tsetlin's results [20] 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 $(M_0 - M^*)$.

We thus conclude that in general, the DL_{RP} automaton performs better in most non-stationary environments 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 $(M_0 - M^*)$ obtained from the DL_{RP} is **many times higher** than the $(M_0 - M^*)$ obtained by using the $L_{2N,2}$ automaton.

VII. CONCLUSIONS AND OPEN PROBLEMS

In this paper we have stated and proved asymptotic results concerning various variable structure stochastic automata. These automata however, unlike most automata discussed in the literature, change the action probabilities in discrete jumps. The automata are called linear because these jumps are all of equal length. We have proved that the DL_{RP} scheme is **ergodic** and is ε-optimal in all environments wherever the minimum penalty probability is less than 0.5. By artificially making the end states of the latter automaton absorbing, we have designed the ADL_{RP} automaton and proven its ε-optimality. This is the only known symmetric Reward-Penalty scheme which is linear and yet ε-optimal. We conjecture that there is none other.

Also by stochastically filtering the inputs to the DL_{RP} automaton we have designed the Modified DL_{RP} (MDL_{RP}) automaton which is the only known **ergodic linear** reward-penalty scheme which is ε-optimal in all random environments.

We are currently investigating the use of these automata to adaptively control a robot manipulator operating in a noisy workspace. The problems of studying nonlinear [14] and multi-action discretized reward-penalty schemes remain open.

ACKNOWLEDGEMENTS

The authors are grateful to David Ng who did much of the simulations for us, and to Robert Cheetham who painstakingly prepared the document from a poorly written manuscript.

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

[4] Lakshmivarahan, 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 Computing Sciences, 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., "Applications 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] Oommen, B.J., "Absorbing and Ergodic Discretized Two-Action Learning Automata", IEEE Trans. on Syst. Man and Cybern., Vol. SMC-16, 1986, pp.282-296.

[15] Oommen, B.J., "A Learning Automaton Solution to the Stochastic Minimum Spanning Circle Problem", IEEE Trans. on Syst. Man and Cybern., July/Aug. 1986, pp.598-603.

[16] Oommen, B.J., and Ma, D.C.Y., "Deterministic Learning Automata Solutions to the Equi-Partitioning Problem". To appear in the IEEE Trans. on Computers (Accepted September 1986).

[17] Oommen, B.J., and Ma, D.C.Y., "Fast Object Partitioning Using Stochastic Learning Automata". Proceedings of the 1987 International Conference on Research and Development in Information Retrieval, New Orleans, June 1987.

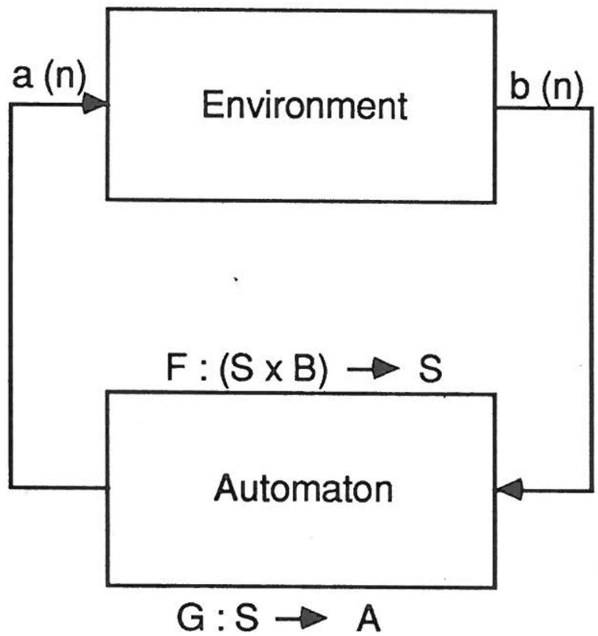
[18] Thathachar, M.A.L., and Oommen, B.J., "Discretized Reward-Inaction Learning Automata", Journal of Cybernetics and Information Sciences, Spring 1979, pp.24-29.

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

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

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

[22] 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.



$$b(n) \in \{0, 1\} = B$$
$$s(n) \in \{s_1, s_2, \ldots, s_N\} = S$$
$$a(n) \in \{a_1, a_2, \ldots, a_R\} = A$$

Figure I : The Automaton-Environment Interaction

Notation : N even

$g_k = k/N$

$g'_k = 1 - k/N$

$d_i = 1 - c_i$; $i = 1, 2.$

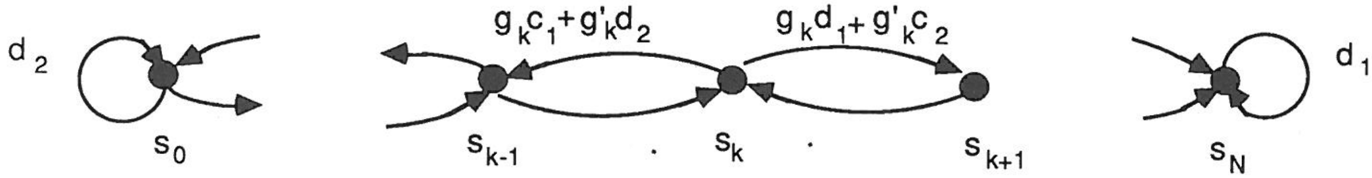


Figure II : The Transition Function of the DL_{RP} Automaton.

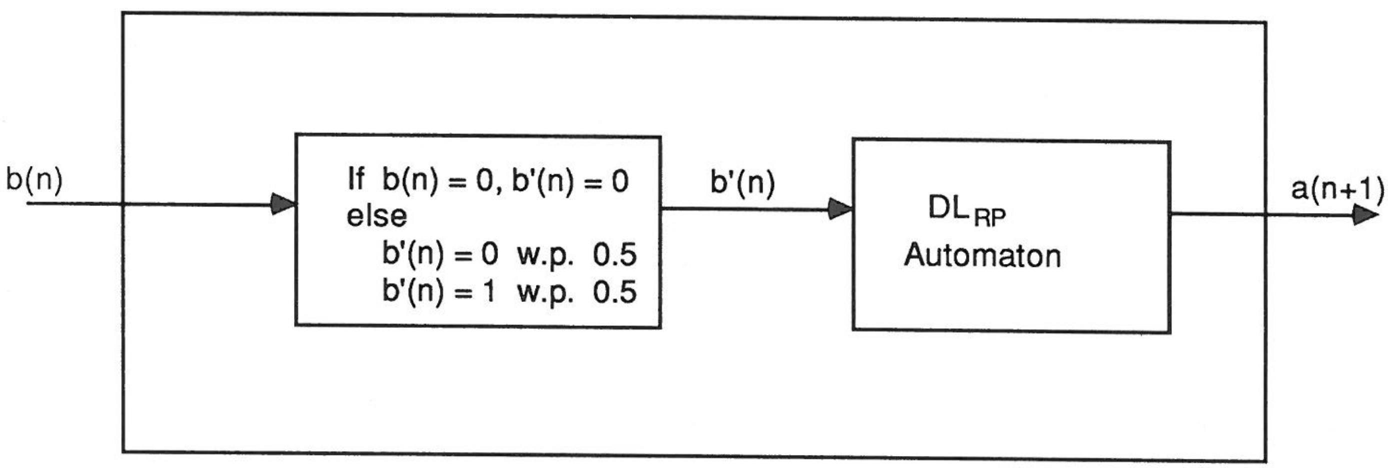


Figure III : The Modified Discretized Linear Reward-Penalty (MDL_{RP}) Automaton viewed as a filter preceding a DL_{RP} Automaton.

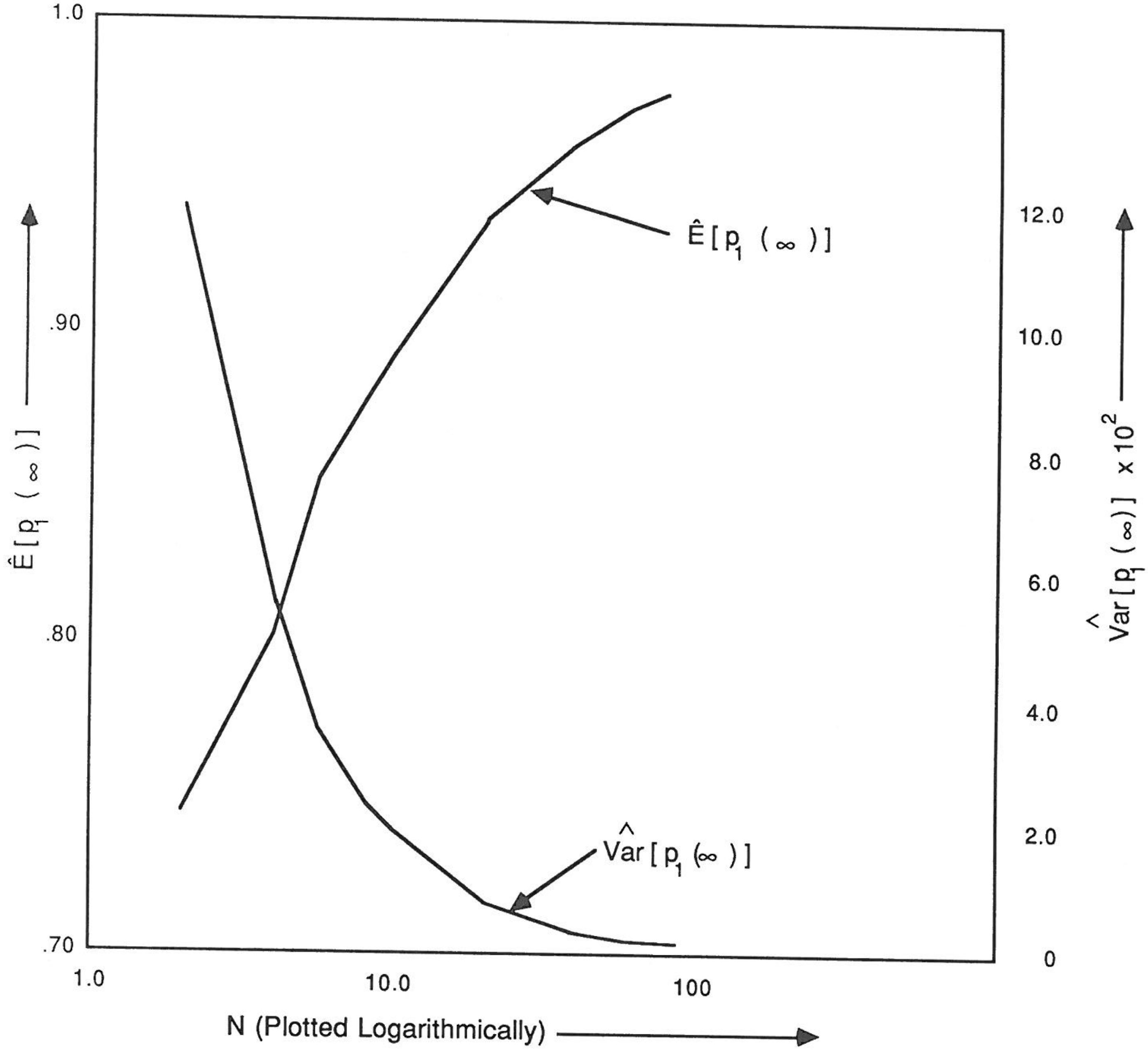


Figure IV : Variation of $\hat{E}[p_1(\infty)]$ and $\hat{Var}[p_1(\infty)]$ with N for the DL_{RP} Automaton. In this case $c_1 = 0.4$ and $c_2 = 0.8$.

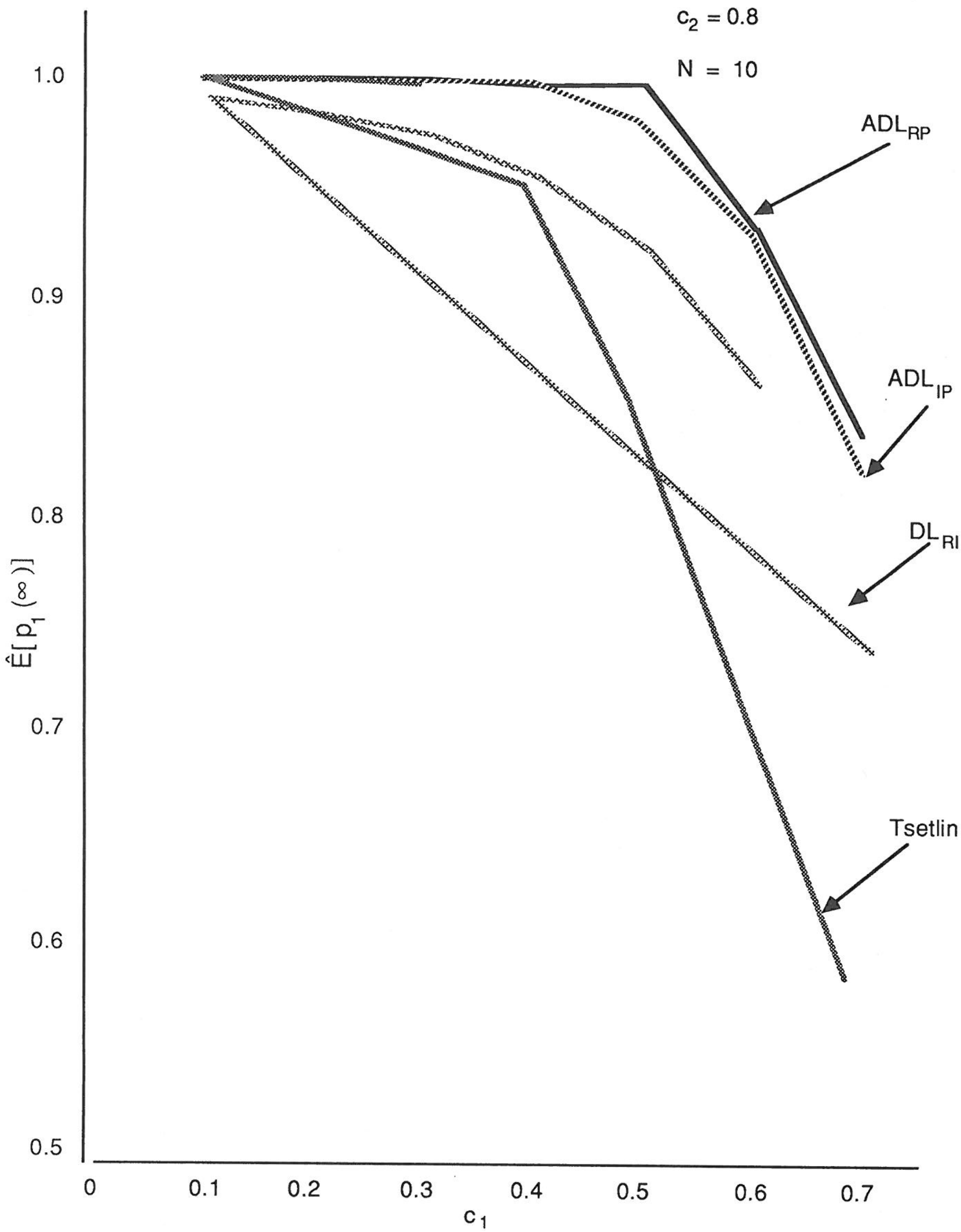


Figure V : A relative comparison of the Tsetlin Automaton, the DL_{RI} scheme, the ADL_{IP} scheme, and the ADL_{RP} scheme for N = 10.

Carleton University, School of Computer Science
Bibliography of Technical Reports
Publications List

School of Computer Science
Carleton University
Ottawa, Ontario, Canada
KIS 5B6

SCS-TR-1 ———	**The Design of CP-6 Pascal** Jim des Rivieres and Wilf R. LaLonde, June 1982
SCS-TR-2 ———	**Single Production Elimination in LR(1) PARSERS: A Synthesis** Wilf R. LaLonde, June 1982
SCS-TR-3 ———	**A Flexible Compiler Structure That Allows Dynamic Phase Ordering** Wilf R. LaLonde and Jim des Rivieres, June 1982
SCS-TR-4 ———	**A Practical Longest Common Subsequence Algorithm for Text Collation** Jim des Rivieres, June 1982
SCS-TR-5 ———	**A School Bus Routing and Scheduling Problem** Wolfgang Lindenberg, Frantisek Fiala, July 1982
SCS-TR-6 ———	**Routing Without Routing Tables: Labelling and Implicit Routing in Networks** Nicola Santoro and Ramez Khatib, July 1982
SCS-TR-7 Out-of-print	**Concurrency Control in Large Computer Networks** Nicola Santoro and Jeffrey B. Sidney, July 1982
SCS-TR-8 Out-of-print	**Order Statistics on Distributed Sets** Nicola Santoro and Jeffrey B. Sidney, July 1982
SCS-TR-9 ———	**Oligarchical Control of Distributed Processing Systems** Moshe Krieger and Nicola Santoro, August 1982
SCS-TR-10 ———	**Communication Bounds for Selection in Distributed Sets** Nicola Santoro and Jeffrey B. Sidney, September 1982
SCS-TR-11 ———	**A Simple Technique for Converting from a Pascal Shop to a C Shop** Wilf R. LaLonde and John R. Pugh, November 1982
SCS-TR-12 ———	**Efficient Abstract Implementations for Relational Data Structures** Nicola Santoro, December 1982
SCS-TR-13 ———	**On The Message Complexity of Distributed Problems** Nicola Santoro, December 1982
SCS-TR-14 out-of-print	**A Common Basis for Similarity Measures Involving Two Strings** R.L. Kashyap and B.J. Oommen, January 1983. See International Journal of Computer Mathematics, March 1983, pp. 17-40.
SCS-TR-15 out-of-print	**Similarity Measures for Sets of String** R.L. Kashyap and B.J. Oommen, January 1983. See International Journal of Computer Mathematics, May 1983, pp. 95-104.

Carleton University, School of Computer Science
Bibliography of Technical Reports

SCS-TR-16 out-of-print
The Noisy Substring Matching Problem
R.L. Kashyap and B.J. Oommen, January 1983. See IEEE Trans. on Software Engg, May 1983, pp. 365-370.

SCS-TR-17 ________
Distributed Election in a Circle Without a Global Sense of Orientation
E. Korach, D. Rotem, and N. Santoro, January 1983

SCS-TR-18 ________
A Geometrical Approach to Polygonal Dissimilarity and the Classification of Closed Boundaries
R.L. Kashyap and B.J. Oommen, January 1983

SCS-TR-19 out-of-print
Scale Preserving Smoothing of Polygons
R.L. Kashyap and B.J. Oommen, January 1983. See IEEE Trans. on Pattern Analysis and Machine Intelligence, Nov. 1983, pp. 667-671.

SCS-TR-20 ________
Not-Quite-Linear Random Access Memories
Jim des Rivieres, Wilf R. LaLonde, and Mike Dixon, March 1983

SCS-TR-21 ________
Shout Echo Selection in Distributed Files
D. Rotem, N. Santoro, J.B. Sydney, March I983.

SCS-TR-22 ________
Distributed Ranking
E. Korach, D. Rotem, N. Santoro, March I983.

SCS-TR-23
A Reduction Technique for Selection in Distributed Files : I
N. Santoro, J.B. Sidney, April I983.
Replaced by SCS-TR-69

SCS-TR-24 out-of-print
Learning Automata Possessing Ergodicity of the Mean : The Two Action Case
M.A.L. Thathachar and B.J. Oommen, May I983. See IEEE Trans. on Systems, Man and Cybernetics, Nov./Dec. 1983, pp. 1143-1148.

SCS-TR-25 ________
Actors - The Stage is Set
John R. Pugh, June I983.

SCS-TR-26 ________
On the Essential Equivalence of Two Families of Learning Automata
M.A.L. Thathachar and B.J. Oommen, May I983.

SCS-TR-27 ________
Generalized Krylov Automata and their Applicability to Learning in Nonstationary Environments
B.J. Oommen, June I983.

SCS-TR-28 out-of-print
Actor Systems: Selected Features
Wilf R. LaLonde, July I983.

SCS-TR-29 ________
Another Addendum to Kronecker's Theory of Pencils
M.D. Atkinson, August I983.

SCS-TR-30 ________
Some Techniques for Group Character Reduction
M.D. Atkinson and R.A. Hassan, August I983.

SCS-TR-3I ________
An Optimal Algorithm for Geometrical Congruence
M.D. Atkinson, August I983.

Carleton University, School of Computer Science
Bibliography of Technical Reports

SCS-TR-32 out-of-print	**Multi-Action Learning Automata Possessing Ergodicity of the Mean** B.J. Oommen and M.A.L. Thathachar, August l983. See Information Science, Vol. 35, No. 3, June 1985, pp. 183-198.
SCS-TR-33 Out-of-print	**Fibonacci Graphs, Cyclic Permutations and Extremal Points** N. Santoro and J. Urrutia, December l983
SCS-TR-34 ________	**Distributed Sorting** D. Rotem, N. Santoro, and J.B. Sydney, December l983.
SCS-TR-35 ________	**A Reduction Technique for Selection in Distributed Files: II** N. Sanatoro, M. Scheutzow, and J.B. Sidney, December l983.
SCS-TR-36 out-of-print	**The Asymptotic Optimality of Discretized Linear Reward-Inaction Learning Automata** B.J. Oommen and Eldon Hansen, January l984. See IEEE Trans. on Systems, Math and Cybernetics, May/June 1984, pp. 542-545.
SCS-TR-37 ________	**Geometric Containment is Not Reducible to Pareto Dominance** N. Santoro, J.B. Sidney, S.J. Sidney, and J. Urrutia, January l984.
SCS-TR-38 ________	**An Improved Algorithm for Boolean Matrix Multiplication** N. Santoro and J. Urrutia, January l984.
SCS-TR-39 ________	**Containment of Elementary Geometric Objects** J. Sack, N. Santoro and J. Urrutia, February l984
SCS-TR-40 ________	**SADE: A Programming Environment for Designing and Testing Systolic Algorithms** J.P. Corriveau and N. Santoro, February l984.
SCS-TR-4l ________	**Intersection Graphs, $\{B_1\}$-Orientable Graphs and Proper Circular Arc Graphs** Jorge Urrutia, February l984.
SCS-TR-42 ________	**Minimum Decompositions of Polygonal Objects** J. Mark Keil and Jörg-R. Sack, March l984.
SCS-TR-43 ________	**An Algorithm for Merging Heaps** Jörg-R. Sack and Thomas Strothotte, March l984.
SCS-TR-44 ________	**A Digital Hashing Scheme for Dynamic Multiattribute Files** E.J. Otoo, March l984
SCS-TR-45 ________	**Symmetric Index Maintenance Using Multidimensional Linear Hashing** E.J. Otoo, March l984
SCS-TR-46 ________	**A Mapping Function for the Directory of a Multidimensional Extendible Hashing** E.J. Otoo, March l984.
SCS-TR-47 ________	**Translating Polygons in the Plane** Jörg-R. Sack, March l984.
SCS-TR-48 ________	**Constrained String Editing** J. Oommen, May l984.

Carleton University, School of Computer Science
Bibliography of Technical Reports

SCS-TR-49 Out-of-print	**O(N) Election Algorithms in Complete Networks with Global Sense of Orientation** Jörg-R. Sack, Nicola Santoro, Jorge Urrutia, May l984
SCS-TR-50 ________	**The Design of a Program Editor Based on Constraints** Christopher A. Carter and Wilf R. LaLonde, May l984.
SCS-TR-5l out-of-print	**Discretized Linear Inaction-Penalty Learning Automata** B.J. Oommen and Eldon Hansen, May l984. Results contained in IEEE Trans. on Systems, Man and Cybernetics, March/April 1986, pp. 292-293.
SCS-TR-52 ________	**Sense of Direction, Topological Awareness and Communication Complexity** Nicola Santoro, May l984.
SCS-TR-53 ________	**Optimal List Organizing Strategy Which Uses Stochastic Move-to-Front Operations** B.J. Oommen, June l984.
SCS-TR-54 ________	**Rectilinear Computational Geometry** J. Sack, June l984.
SCS-TR-55 ________	**An Efficient, Implicit Double-Ended Priority Queue** M.D. Atkinson, Jörg-R. Sack, Nicola Santoro, T. Strothotte, July l984.
SCS-TR-56 ________	**Dynamic Multipaging: A Multidimensional Structure for Fast Associative Searching** E. Otoo, T.H. Merrett, August l984.
SCS-TR-57 ________	**Specialization, Generalization and Inheritance** Wilf R. LaLonde, John R. Pugh, August l984.
SCS-TR-58 ________	**Computer Access Methods for Extendible Arrays of Varying Dimensions** E. Otoo, August l984.
SCS-TR-59 ________	**Area-Efficient Embeddings of Trees** J.P. Corriveau, Nicola Santoro, August l984.
SCS-TR-60 ________	**Uniquely Colourable m-Dichromatic Oriented Graphs** V. Neumann-Lara, N. Santoro, J. Urrutia, August l984.
SCS-TR-6l ________	**Analysis of Distributed Algorithms for Extrema Finding in a Ring** D. Rotem, E, Korach and N. Santoro, August l984.
SCS-TR-62 out-of-print	**On Zigzag Permutations and Comparisons of Adjacent Elements** M.D. Atkinson, October l984. See Information Processing Letters 21 ('85) 187-189
SCS-TR-63 ________	**Sets of Integers with Distinct Differences** M.D. Atkinson, A. Hassenklover, October l984.
SCS-TR-64 ________	**Teaching Fifth Generation Computing: The Importance of Small Talk** Wilf R. LaLonde, Dave A. Thomas, John R. Pugh, October l984.
SCS-TR-65 ________	**An Extremely Fast Minimum Spanning Circle Algorithm** B.J. Oommen, October l984.

SCS-TR-66 **On the Futility of Arbitrarily Increasing Memory Capabilities of Stochastic Learning Automata**
B.J. Oommen, October 1984. Revised May 1985.

SCS-TR-67 **Heaps in Heaps**
T. Strothotte, J.-R. Sack, November 1984. Revised April 1985.

SCS-TR-68 out-of-print **Partial Orders and Comparison Problems**
M.D. Atkinson, November 1984. See Congressus Numerantium 47 ('86), 77-88

SCS-TR-69 **On the Expected Communication Complexity of Distributed Selection**
N. Santoro, J.B. Sidney, S.J. Sidney, February 1985.

SCS-TR-70 **Features of Fifth Generation Languages: A Panoramic View**
Wilf R. LaLonde, John R. Pugh, March 1985.

SCS-TR-71 **Actra: The Design of an Industrial Fifth Generation Smalltalk System**
David A. Thomas, Wilf R. LaLonde, April 1985.

SCS-TR-72 **Minmaxheaps, Orderstatisticstrees and their Application to the Coursemarks Problem**
M.D. Atkinson, J.-R. Sack, N. Santoro, T. Strothotte, March 1985.

SCS-TR-73 **Designing Communities of Data Types**
Wilf R. LaLonde, May 1985.
Replaced by SCS-TR-108

SCS-TR-74 out-of-print **Absorbing and Ergodic Discretized Two Action Learning Automata**
B. John Oommen, May 1985. See IEEE Trans. on Systems, Man and Cybernetics, March/April 1986, pp. 282-293.

SCS-TR-75 **Optimal Parallel Merging Without Memory Conflicts**
Selim Akl and Nicola Santoro, May 1985

SCS-TR-76 **List Organizing Strategies Using Stochastic Move-to-Front and Stochastic Move-to-Rear Operations**
B. John Oommen, May 1985.

SCS-TR-77 **Linearizing the Directory Growth in Order Preserving Extendible Hashing**
E.J. Otoo, July 1985.

SCS-TR-78 **Improving Semijoin Evaluation in Distributed Query Processing**
E.J. Otoo, N. Santoro, D. Rotem, July 1985.

SCS-TR-79 **On the Problem of Translating an Elliptic Object Through a Workspace of Elliptic Obstacles**
B.J. Oommen, I. Reichstein, July 1985.

SCS-TR-80 **Smalltalk - Discovering the System**
W. LaLonde, J. Pugh, D. Thomas, October 1985.

SCS-TR-81 **A Learning Automation Solution to the Stochastic Minimum Spanning Circle Problem**
B.J. Oommen, October 1985.

Carleton University, School of Computer Science
Bibliography of Technical Reports

SCS-TR-82 **Separability of Sets of Polygons**
Frank Dehne, Jörg-R. Sack, October 1985.

SCS-TR-83 out-of-print **Extensions of Partial Orders of Bounded Width**
M.D. Atkinson and H.W. Chang, November 1985. See Congressus Numerantium, Vol. 52 (May 1986), pp. 21-35.

SCS-TR-84 **Deterministic Learning Automata Solutions to the Object Partitioning Problem**
B. John Oommen, D.C.Y. Ma, November 1985

SCS-TR-85 out-of-print **Selecting Subsets of the Correct Density**
M.D. Atkinson, December 1985. To appear in Congressus Numerantium, Proceedings of the 1986 South-Eastern conference on Graph theory, combinatorics and Computing.

SCS-TR-86 **Robot Navigation in Unknown Terrains Using Learned Visiblity Graphs. Part I: The Disjoint Convex Obstacles Case**
B. J. Oommen, S.S. Iyengar, S.V.N. Rao, R.L. Kashyap, February 1986

SCS-TR-87 **Breaking Symmetry in Synchronous Networks**
Greg N. Frederickson, Nicola Santoro, April 1986

SCS-TR-88 **Data Structures and Data Types: An Object-Oriented Approach**
John R. Pugh, Wilf R. LaLonde and David A. Thomas, April 1986

SCS-TR-89 **Ergodic Learning Automata Capable of Incorporating Apriori Information**
B. J. Oommen, May 1986

SCS-TR-90 **Iterative Decomposition of Digital Systems and Its Applications**
Vaclav Dvorak, May 1986.

SCS-TR-91 **Actors in a Smalltalk Multiprocessor: A Case for Limited Parallelism**
Wilf R. LaLonde, Dave A. Thomas and John R. Pugh, May 1986

SCS-TR-92 **ACTRA - A Multitasking/Multiprocessing Smalltalk**
David A. Thomas, Wilf R. LaLonde, and John R. Pugh, May 1986

SCS-TR-93 **Why Exemplars are Better Than Classes**
Wilf R. LaLonde, May 1986

SCS-TR-94 **An Exemplar Based Smalltalk**
Wilf R. LaLonde, Dave A. Thomas and John R. Pugh, May 1986

SCS-TR-95 **Recognition of Noisy Subsequences Using Constrained Edit Distances**
B. John Oommen, June 1986

SCS-TR-96 **Guessing Games and Distributed Computations in Synchronous Networks**
J. van Leeuwen, N. Santoro, J. Urrutia and S. Zaks, June 1986.

SCS-TR-97 **Bit vs. Time Tradeoffs for Distributed Elections in Synchronous Rings**
M. Overmars and N. Santoro, June 1986.

SCS-TR-98 **Reduction Techniques for Distributed Selection**
N. Santoro and E. Suen, June 1986.

Carleton University, School of Computer Science
Bibliography of Technical Reports

SCS-TR-99 **A Note on Lower Bounds for Min-Max Heaps**
A. Hasham and J.-R. Sack, June 1986.

SCS-TR-100 **Sums of Lexicographically Ordered Sets**
M.D. Atkinson, A. Negro, and N. Santoro, May 1987.

SCS-TR-102 **Computing on a Systolic Screen: Hulls, Contours, and Applications**
F. Dehne, J.-R. Sack and N. Santoro, October 1986.

SCS-TR-103 **Stochastic Automata Solutions to the Object Partitioning Problem**
B.J. Oommen and D.C.Y. Ma, November 1986.

SCS-TR-104 **Parallel Computational Geometry and Clustering Methods**
F. Dehne, December 1986.

SCS-TR-105 **On Adding *Constraint Accumulation* to Prolog**
Wilf R. LaLonde, January 1987.

SCS-TR-107 **On the Problem of Multiple Mobile Robots Cluttering a Workspace**
B. J. Oommen and I. Reichstein, January 1987.

SCS-TR-108 **Designing Families of Data Types Using Exemplars**
Wilf R. LaLonde, February 1987.

SCS-TR-109 **From Rings to Complete Graphs - $\Theta(n \log n)$ to $\Theta(n)$ Distributed Leader Election**
Hagit Attiya, Nicola Santoro and Shmuel Zaks, March 1987.

SCS-TR-110 **A Transputer Based Adaptable Pipeline**
Anirban Basu, March 1987.

SCS-TR-111 **Impact of Prediction Accuracy on the Performance of a Pipeline Computer**
Anirban Basu, March 1987.

SCS-TR-112 **ε-Optimal Discretized Linear Reward-Penalty Learning Automata**
B.J. Oommen and J.P.R. Christensen, May 1987.