A LEARNING AUTOMATON SOLUTION TO THE STOCHASTIC MINIMUM SPANNING CIRCLE PROBLEM

B. John Oommen

SCS-TR-81

October 1985

School of Compueter Science
Carleton University
Ottawa, Ontario
Canada K1S 5B6

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

A LEARNING AUTOMATON SOLUTION TO THE STOCHASTIC MINIMUM SPANNING CIRCLE PROBLEM*

B. John Oommen+

ABSTRACT

The Minimum Spanning Circle (MSC) of N points in the plane is the smallest circle that encloses these points. This problem has been extensively studied in the literature. [1,2,4,10,11]. In this paper we consider the problem of computing the MSC of N stochastically varying points in the plane. We propose a solution to the problem which involves a heirarchy of learning automata. The automata used in this solution are the Absorbing Discretized Linear Inaction-Penalty (ADL_{IP}) automata which are the only known linear automata which are of an inaction-penalty flavour and yet asymptotically optimal.

---

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

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

I. INTRODUCTION

In the areas of robotics, image analysis and computer vision the problem of geometric modelling has come to the limelight in the recent past. Informally stated, this problem involves fitting and manipulating a geometric model for a set of points in two or three dimensions. The simplest model for a closed convex figure described by N points is the convex hull of these points. However, in many situations it is easier to work with a non-polygonal model of these points described algebraically. And in this context, the simplest geometric model for the convex polygon is the circle.

The Minimum Spanning Circle (MSC) of a set of N points in the plane is the smallest (unique) circle which encloses all these points. The MSC problem was first presented by Sylvester in 1857 [12]. Various algorithms have been proposed since then [1,2,4,10,11]. For the sake of brevity we refer the reader to an excellent paper by Hearn and Vijay [2] which surveys the literature and qualitatively and computationally compares the various algorithms which solve the problem. Two solutions not cited in [2] are the ones due to Shamos [11] and Megiddo [4]. Shamos' algorithm involved the Fartherest Point Voronoi Diagram and was shown to be of complexity O(N.log N). Shamos conjectured that this was a lower bound on any algorithm that solved the problem. Recently however, by transforming the problem to be a linear programming problem in two dimensions, Megiddo disproved Shamos' conjecture by proposing a linear time solution for the MSC problem. More recently [10], we have presented an algorithm which is a geometric construction, and which converges extremely fast. In every single experiment conducted involving both random and digital data (without even a rare exception) the algorithm has converged in exactly **two** iterations. We conjecture that the algorithm described in [10] has a linear time conplexity.

When the points involved are deterministic, any of the above algorithms can be used to compute their MSC. However, if the points themselves are unobservable, but noisy versions of these points are observable (for example, by processing a digital image), the problem becomes a lot more complex. This paper concerns the computation of the MSC of the points in such a case. Explicitly stated the problem which we study in this paper involves computing the MSC of a set $\{X_i | 1 \leq i \leq N\}$ by processing a sequence of noisy observations $\{Y_i(t) \mid 1 \leq i \leq N; t \geq 0\}$, where for all i,

$$Y_i(t) = X_i + \eta, \text{ with}$$
$$E[\eta] = 0, \text{ and } Var[\eta] < \infty.$$

We assume that the random noise η is independent and identically distributed. We shall refer to this problem as the Stochastic MSC problem, and shall propose the first known solution to the problem.

I.1. Outline of Solution Proposed

The solution which we propose involves using a heirarchy of discretized learning automata. Automata models of learning systems 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. Automata which possess learning capabilities have been used in various applications [3] including game playing, pattern recognition and hypothesis testing, priority assignment in a queueing system and telephone routing [5-7]. The literature on learning automata is extensive. We refer the reader to a review paper by Narendra and Thathachar [5] and a recent excellent book by Lakshmivarahan [3] for a review of the families and applications of learning automata.

A Variable Structure Stochastic Automaton (VSSA) is a learning machine that is completely defined by a set of probability updating rules. The automaton chooses one of the actions offered by the environment based on its action probability vector $\underline{p}(t)$. Based on the response it receive from the environment, the automaton updates the vector $\underline{p}(t)$ and chooses its action at the next time instant. The VSSA that are used in this paper are called the Absorbing Discretized Linear Inaction-Penalty (ADL_{IP}) automata. Although the structure of this automaton is left to a later section, we shall remark that this automaton was recently discovered by us [8,9] to be an extremely accurate automaton. Further, it is the only known linear machine which possesses an inaction-penalty flavour and which is simultaneously asymptotically optimal. Of all the learning automata studied in the literature, we believe that this automaton is most suited for problems in which the penalty probabilities associated with the actions to be chosen are extremely high. This has been the primary motivation for us to use the ADL_{IP} automaton to tackle the Stochastic MSC problem, because, as we shall see later, in the latter problem the reward probabilities are so insignificant, and hence a learning machine of any other flavour would be inadequate.

The organization of the paper is as follows. We shall first describe the ADL_{IP} automata and state some powerful results concerning its asymptotic optimality. We shall then describe the heirarchy of ADL_{IP} automaton which is used to tackle the Stochastic MSC problem. We conclude the paper with some experimental results.

II. The Absorbing Discretized Linear Inaction-Penalty Automaton

II.1. Fundamentals of Learning Automata

The automaton considered in this paper selects an action a(t) at each time instant 't' from a finite action set $\{a_i \mid i = 1 \text{ to } R\}$. The selection is done on the basis of a probability distribution $\underline{p}(t)$, an R x 1 vector where, for all t,

$$p_i(t) = Pr[a(t) = a_i], \text{ with } \Sigma_{1 \leq i \leq R}\, p_i(t) = 1. \qquad (1)$$

Throughout this paper we assume that R = 2.

The selected action serves as the input to the environment which gives out a response b(t) at time 't'. b(t) 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,

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

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

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 is the minimum of the penalty probabilities, then $p_L(t) = 1$, $p_i(t) = 0$ for $i \neq L$ achieves this result. Updating schemes for $\underline{p}(t)$ are to be chosen with this optimal solution in view.

An automaton is said to be **optimal** if the expected penalty equals c_L the minimum penalty probability as time goes to infinity. It is ε–optimal if the limiting expected penalty is less than $c_{min} + \varepsilon$ for some arbitrary $\varepsilon > 0$ by suitable choice of some parameters of the automaton. Thus the limiting value of the expected penalty can be made as close to c_L as desired.

II.2. Introduction to Discretized Automata

Variable Structure Stochastic Automata (VSSA) are implemented using a Random Number Generator. 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 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 requires on the random number generator **and to increase the speed of convergence** of the VSSA the concept of discretizing the probability space was recently introduced in the literature. 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. Subsequently, the two action Discretized Linear Reward-Inaction (DL_{RI}) automaton was proven to be ε-optimal in all random environments. The latter is more accurate and converges faster than its continuous counterpart. Later, in [9] various families of ergodic and absorbing discretized two action automaton were introduced. Among other results proved in [9], a new automaton known as the ADL_{IP} automaton was shown to be ε-optimal. This is the only scheme known to us to be of a linear inaction-penalty nature and simultaneously ε-optimal. For the sake of continuity, this automaton is described here and its asymptotic properties alluded to. However, for the sake of brevity the proofs of these properties are omitted. They are included in [9].

II.3. The ADL_{IP} Automaton

The Absorbing Discretized Linear Inaction-Penalty (ADL_{IP}) Automaton is defined as a pair (S,F) where,

(i) $S = \{s_0, s_1, \ldots, s_M\}$ is the set of states it possesses, and

(ii) F is a map from S x B to S and is called the transition map.
We now describe the properties of S and F respectively.

The set S contains M+1 elements, where M is an **even** integer. Associated with the state s_i is the probability i/M. This represents $p_1(t)$, the probability of the automaton choosing action a_1. Observe that in this state the automaton chooses the action a_2 with the probability (1 - i/M). Since any one of the action probabilities completely defines the vector of action probabilities, we shall, with no loss of generality, consider $p_1(t)$. M is termed as the Depth of Memory of the machine.

G is the state transition map specified by (3) below for $1 \leq i \leq M-1$.

$$\begin{aligned} s(t+1) &= s_{i+1} && \text{if } a(t) = a_2 \text{ and } b(t) = 1, \\ &= s_{i-1} && \text{if } a(t) = a_1 \text{ and } b(t) = 1, \\ &= s_i && \text{if } a(t) = a_1 \text{ or } a_2 \text{ and } b(t) = 0. \end{aligned} \quad (3)$$

Further, s_0 and s_M are absorbing states, and thus, if $s(t) = s_0$ then $s(t+1) = s_0$, and if $s(t) = s_M$, then $s(t+1) = s_M$, for all t. (See Figure I).

Obviously $p_1(t)$ 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 P be the stochastic matrix defining the Markov chain, where the individual element of P is,

$$P_{i,j} = Pr[s(t) = s_j \mid s(t-1) = s_i].$$

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

$$\begin{aligned} P_{i,i+1} &= h_i c_2, \\ P_{i,i-1} &= g_i c_1, \\ P_{i,i} &= 1 - g_i c_1 - h_i c_2, \end{aligned} \quad (4)$$

where $g_i = i/M$ and $h_i = 1 - i/M$.
Besides $P_{0,0} = p_{M,M} = 1$. All the other elements of P are identically m zero. We now consider the asymptotic properties of ADL_{IP} Scheme.

II.4. The ε-optimality of the ADL_{IP} Scheme

Consider the Markov chain which represents the ADL_{IP} automaton. As s_0 and s_M are absorbing states and there are no periodic states, it is clear that $p_1(t)$ tends to 0 or 1 with probability 1. However, only one of these terminal values is desired. For fixing ideas let us assume $c_1 < c_2$. Then it is desired that $p_1(t)$ should tend to 1 and s(t) to s_M. We can prove that this is indeed the case.

Theorem I

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

Sketch of Proof

We intend to prove that $\lim_{M\to\infty} \lim_{t\to\infty} p_1(t) = 1$.

To do this we first prove that for any even M, the probability of converging to s_M is given by

$$H_M = (\Sigma_{0\le i\le M/2-1} R_i) / (\Sigma_{0\le i\le M-1} R_i) \tag{5}$$

where $R_i = \Pi_{1\le j\le i}\ u_j/f_j$ and u_j and f_j are defined by (6) below.

$$f_j = (1 - j/M).c_2 \qquad \text{and} \qquad u_j = (j/M).c_1 \tag{6}$$

Simplifying (5), after some manipulation yields:

$$H_M = \frac{\{\Sigma_{0\le i\le M/2-1}\ (\ (M-1-i)!\ i!/(M-1)!\ ).e^i\}}{\{\Sigma_{0\le i\le M-1}\ (\ (M-1-i)!\ i!/(M-1)!\ ).e^i\}}, \quad \text{where, } e = c_1/c_2. \tag{7}$$

Using the latter the theorem follows since it can be shown that

$$\lim_{M\to\infty} H_M = 1. \tag{8}$$

•••

The individual proofs for showing (5), (7) and (8) are quite involved. They are given in [9], in which are also presented experimental results of the ADL_{IP} automaton interacting with various random environments. These results demonstrate that the ADL_{IP} automaton is extremely accurate.

Apart from pathological importance (namely, the importance of the scheme in that it is the only known ε-optimal linear inaction-penalty automaton) the ADL_{IP} automaton has also a very interesting practical application. It is possibly one of the best automata which can be used to interact with an environment whose penalty probabilities are very high. Suppose the value of c_L, the minimum penalty probability was 0.995. This could imply that on the average, out of a thousand responses from the environment not more than five of them will be a reward. Since a symmetric linear Reward-Penalty scheme is at its best expedient and definitely not ε–optimal such a scheme is rejectable. The L_{RI} (Reward-Inaction) scheme is ε–optimal, but since the reward probabilities are extremely low, the automaton will ignore most of the responses from the environment. What really is needed in this case is a scheme which "squeezes" out the information from the penalty responses and which simultaneously is ε–optimal. We believe that the ADL_{IP} automaton is extremely suited for such applications.This renders it readily applicable for the main problem considered in this paper, namely, the Stochastic MSC problem.

III. The Solution to the Stochastic MSC Problem.

Our aim is to compute the MSC of N points $\{X_i \mid 1 \le i \le N\}$ by processing a sequence $\{Y_i(t) \mid 1 \le i \le N\}$, where,

$Y_i(t) = X_i + \eta$, where

η is i.i.d., with $E[\eta] = 0$ and $Var[\eta] < \infty$.

Note that the points $\{X_i\}$ themselves cannot be observed.

The MSC of a set of deterministic points has the following interesting property: If the MSC touches exactly two of the N points, then these two points must be on the diameter of the circle. However, if the MSC touches exactly k ($k \ge 3$) points, then there exists exactly three points of these k which form a non-obtuse angled triangle. The MSC is the circumcircle of this triangle [1,2,10].

The method by which we have tackled this problem is by using automata to determine the **indices** of the points $\{X_i\}$ which actually determine the MSC. To do this we have used two levels of ADL_{IP} automata. In this first level the automaton decides whether the number of points determining the MSC is two or three. At the second level, there is an automaton for each point deciding whether it is the point defining the MSC or not. Note that this decision is made **dependent** on the fact that either two or three points determines the MSC, since, the probability $Pr\left[X_i \text{ is on MSC} \mid 2 \text{ points determines MSC}\right]$ need not equal the probability $Pr\left[X_i \text{ is on MSC} \mid 3 \text{ points determine MSC}\right]$.

Let us suppose that the MSC is determined by exactly two points X_a and X_b, where a and b are the **unknown** indices sought for. Since any other choice of points will result in an unsuccessful "guess", it will effectively result in "penalizing" the automaton. Since this is true for all other pairs of points, it is clear that the penalty probabilities involved are very high, and so, the reward probabilities are almost insignificant. This is primarily what motivates the use of the ADL_{IP} automaton as the learning machine to solve the problem.

The strategy at the first level is quite straighforward. The automaton decides on the number of points describing the MSC using a 2-action ADL_{IP} automaton whose actions are α_{12} and α_{13}, where:

α_{12} : Decision that 2 points determine the MSC
α_{13} : Decision that 3 points determine the MSC.

At the second level every point has a 2-action ADL_{IP} automaton, the two actions being that of either considering the point or not considering it. Based on these action probabilities (correspondingly normalized) an index 'a' is chosen, $1 \leq a \leq N$.

In what follows, to simplify the notation we shall omit the parameter measuring time. Thus for all i, Y_i with refer to $Y_i(t)$ unless explicitly stated.

If the first level dictates that two points determines the MSC, a second index 'b' is chosen, $1 \leq b \leq N$, where $a \neq b$, and the point Y_b is the most distant point from Y_a. Observe that in this case, if 'a' was a point determining the MSC, then, due to the property of the MSC, the fartherest point from it will indeed be on its diameter. If the circle with Y_a and Y_b as diameter does not span the set $\{Y_i(t)\}$, we reckon that our decision is being penalized. In this case, the strategy involves decrementing the probability of choosing the 2-point rule at the first level. Further, the probability of choosing the index 'a' if the 2-point rule is being used is also decremented.

Similarly, if the 3-point rule is the one being used, the first index 'a' is chosen at random. Using the centroid of $\{Y_i(t)\}$ as the center of a hypothetical circle, a point Z_a, diametrically opposite to Y_a is located. Note that in general Z_a need not be a member of $\{Y_i(t)\}$. The second index, 'b' in this case is the index of the point Y_k which minimizes the angle $\angle Y_a Y_k Z_a$. Using the points Y_a and Y_b, the third point Y_c is determined as the point Y_j which minimizes the angle $\angle Y_aY_jY_b$. If the circumcircle of $Y_aY_bY_c$ does not span the set $\{Y_i(t)\}$, or if the angle $\angle Y_aY_cY_b$ is obtuse, our decision is penalized. Hence the strategy decrements the probability of choosing the 3-point rule. Besides the probability of choosing the index 'a' if the 3-point rule is being used is also decremented. The reasoning for the above strategy is a consequence of the properties of the MSC which have been used in various algorithm discussed in the literature [1,2,10].

Although, in concept, we maintain a ADL_{IP} automaton at the first level and ADL_{IP} automata for each point at the second level, in practice, all that this involves is the maintanance of an integer memory location which remembers which state the corresponding ADL_{IP} automaton is in. On being penalized this memory location is decremented. Indeed all that this entails is the maintainence and updating of exactly 2N+1 integer memory locations. For the sake of completeness we now present the above procedure algorithmically.

ALGORITHM StochasticMSC

Input: A sequence of points $\{Y_i(t)\}$, where for all i, $Y_i(t) = X_i + \eta$, and η is an unknown i.i.d. noise with $E[\eta] = 0$, $Var[\eta] < \infty$.

Output: The indices of the points $\{X_i\}$ which describe the MSC of $\{X_i\}$.

Assumption: The user can exit from the algorithm whenever the vector of total probabilities is almost unchanged. In this case, the Boolean variable Satisfied is set to TRUE.

NOTATION: (i) M is the memory associated with each machine. Hence probability increments and decrements are of magnitude 1/M.

(ii) q is the probability of deciding on the 2-point rule. 1-q is the probability of deciding on the 3-point rule.

(iii) w_{2i} is the probability that point i will be chosen given that the 2-point rule is chosen. Similarly, $1-w_{2i}$ is the corresponding probability that point i will not be chosen. Analogously, the quantities w_{3i} and $1-w_{3i}$ defines the probability of choosing or not choosing the index i given the 3-point rule.

```
METHOD: Satisfied = FALSE          (* Initialize *)
        q = 0.5
        FOR i = 1 TO N DO
            w_2i = w_3i = 0.5
        ENDFOR
        REPEAT UNTIL (Satisfied)
            Read input point {Y_i}
            J = RuleChosen (*Choose 2-point or 3-point Rule based on q *)
            a = FirstPointChosen (*Choose first point base on the w_Ji 's *)
            IF (J = 2) THEN
                b = Index of point fartherest from Y_a
                IF ( Circle with Y_aY_b as diameter does not span {Y_i} ) THEN
                    q = q - (1/M)
                    w_2a = w_2a - (1/M)
                ENDIF
            ELSE (* J must have value 3 *)
                Θ = Centroid of {Y_i}
                Z_a = Point diametrically opposite Y_a on the line Y_aΘ
                b = Index of point minimizing angle ∠ Y_aY_iZ_a
                c = Index of point minimizing angle ∠ Y_aY_iY_b
                IF ( Circumcircle of Y_aY_bY_c does not span {Y_i}
                            OR ∠Y_aY_cY_b > π/2 ) THEN
                    q = q + (1/M )  (* Decrement probability of 3-point Rule *)
                    w_3a = w_3a - (1/M)
                ENDIF
            ENDIF
            Recompute (Satisfied)
        ENDREPEAT
END Algorithm StochasticMSC.
```

III.1. Experimental Results

The algorithm which we have proposed has been tested for a **variety** of data sets. To simulate the random environment the set $\{X_i\}$ was assumed and using a gaussian random number generator the sequence $\{Y_i(t)\}$ was obtained. This sequence of points served as the input to the algorithm. For each data set, 10 experiments were run and in each run 1000 iterations were carried out. For the sake of brevity, we report here the results of **one** such data set using two different noise levels.

The data set consisted of 12 points. These points are shown in Figure II.(a). Around these points are shown small circles from which the noisy versions of the original points came. The small circles represent a variation of three standard deviations. Notice that in this case the MSC of $\{X_i\}$ is determined by exactly two points (say X_a and X_b). The variation of the average probability of the 2-point and the 3-point rule are shown in Figure II(b). In Figure II(c), the variation of the average conditional probability of choosing point X_a and any other point X_i $(i \neq a,b)$ is shown. Observe that generally speaking, the average probability of choosing the 2-point rule increases monotonically from its initial value of 0.5 to its final value. The average conditional probability of choosing X_a increases in a similar way, and the corresponding probability of choosing any point X_i $(i \neq a,b)$ tends to decrease monotonically..

The same set of points is shown in Figure III. Notice that in this case the noise level is **much** higher. In this case too, Figure III(b) shows the variation of the average probability of the 2-point rule and the 3-point rule being chosen. Figure III(c) depicts the variation of the average conditional probability of choosing point X_a and any other point X_i, $(i \neq a,b)$. The power of the technique is obvious.

These graphs are typical of the results that we have obtained. We summarize the characterstics of the algorithm below:

(i) If the MSC of $\{X_i\}$ is well defined by exactly 2 or exactly 3 points, and the noise level is low, the algorithm converges "fairly quickly". The rate of convergence decreases in this case as the noise level increases.

(ii) If the MSC of $\{X_i\}$ is defined by both 2 **and** 3 points (i.e., the 3 points define a right angled triangle), the algorithm tends to converge to the 3-point rule.

(iii) Although the ADL_{IP} is **very** accurate, the automaton itself, in very difficult environments, can be rather sluggish [9]. The algorithm StochasticMSC tends to exhibit a similar characterstics. If there is a faster ε-optimal Inaction-Penalty automaton, we would recommend the use of such automata in a configuration identical to the one we have described in this paper. However, from the nature of the problem it seems obvious that an **Inaction-Penalty** automaton is the one to be sought for.

(iv) By using a deterministic MSC algorithm for the first few iterations, the probabilities q, $\{w_{2i}\}$ and $\{w_{3i}\}$ can be initialized to values other than 0.5. This would reflect our "subjective" a priori feeling concerning the solution, based on the solution obtained after these initial iterations. This can be used to hasten the convergence. An example of this is given in Figure IV for the same data and noise as described in Figure II.

IV CONCLUSIONS

In this paper we have discussed the problem of finding the MSC of an unobservable set $\{X_i\}$. This is done by processing a sequence of noisy observations of $\{X_i\}$. At every time instant a set $\{Y_i(t)\}$ serves as the input. We have proposed a solution to the problem using a two level heirarchy of learning automata. However, as opposed to the learning automata discussed in the literature, we have used an automaton known as the ADL_{IP} scheme [9]. This is the only known linear scheme which is of an inaction-penalty flavour and which is also ε-optimal.

The algorithm which we have proposed is the only known solution to the problem. Experimental results involving various data sets and noise levels show that the algorithm is accurate, but rather sluggish. We suggest that the rate of convergence can be improved if any other ε-optimal Inaction-Penalty automaton faster than the ADL_{IP} scheme is used in a configuration similar to the one described here.

REFERENCES

[1] Chrystal, G., "On the Problem to Construct the Minimum Circle Enclosing n Given Points in the Plane", Proc. Edinburgh Math. Soc. 3, 1885, pp. 30-33.

[2] Hearn, D.W., and Vijay, J., "Efficient Algorithms for the (Weighted) Minimum Circle Problem", Operations Research, 30, 1982, pp.777-795.

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

[4] Megiddo, N., "Linear Time Algorithms for Linear Programming in R X R X R and Related Problems", Proceeding of the 23rd Annual Symposium on the Foundations of Computer Science, 1982, pp. 151-162.

[5] 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.

[6] 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.

[7] 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.

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

[9] Oommen, B.J., "Absorbing and Ergodic Discretized Two Action Learning Automata", To appear in IEEE Trans. on Systems, Man and Cybernetics.

[10] Oommen, B.J., "A Possibly Linear Minimum Spanning Circle Algorithm", Proceedings of the 1985 Allerton Conference, Urbana, Illinois. Also available as the SCS-TR-65, from School of Computer Science, Carleton University, Ottawa, K1S 5B6.

[11] Shamos, M.I., Computational Geometry, Ph. D. Thesis, Yale University, 1978.

[12] Sylvester, J.J., "A Question in the Geometry of Situation", Quart. J. Pure Appl. Math. 1, 1857, p.79.

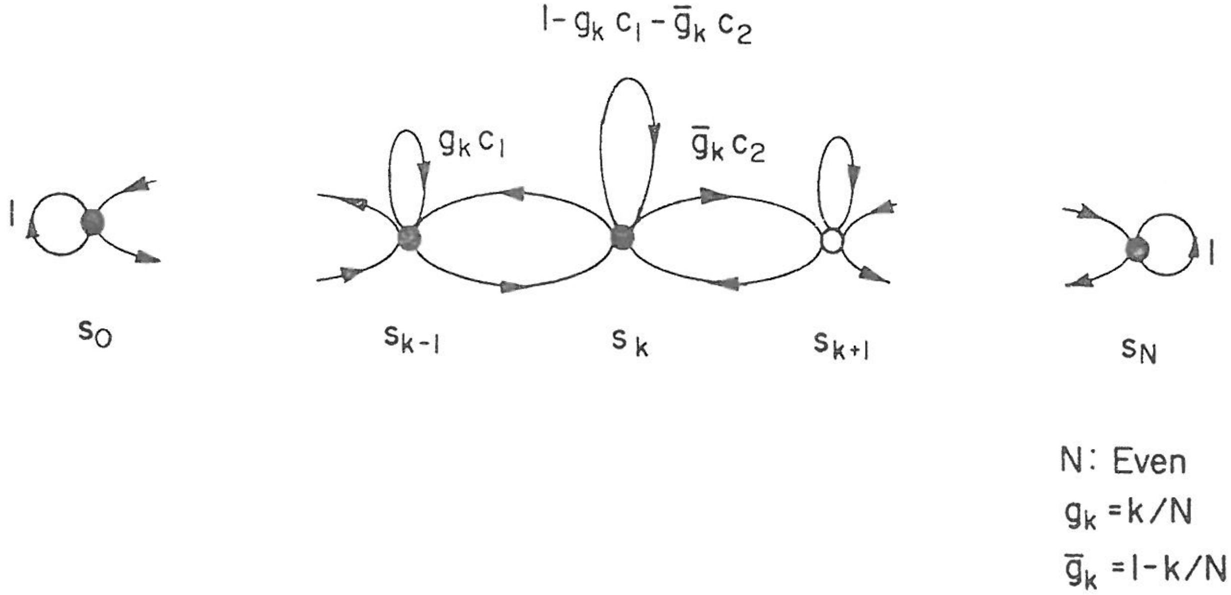

FIG. 1 : THE ADL_{IP} AUTOMATON

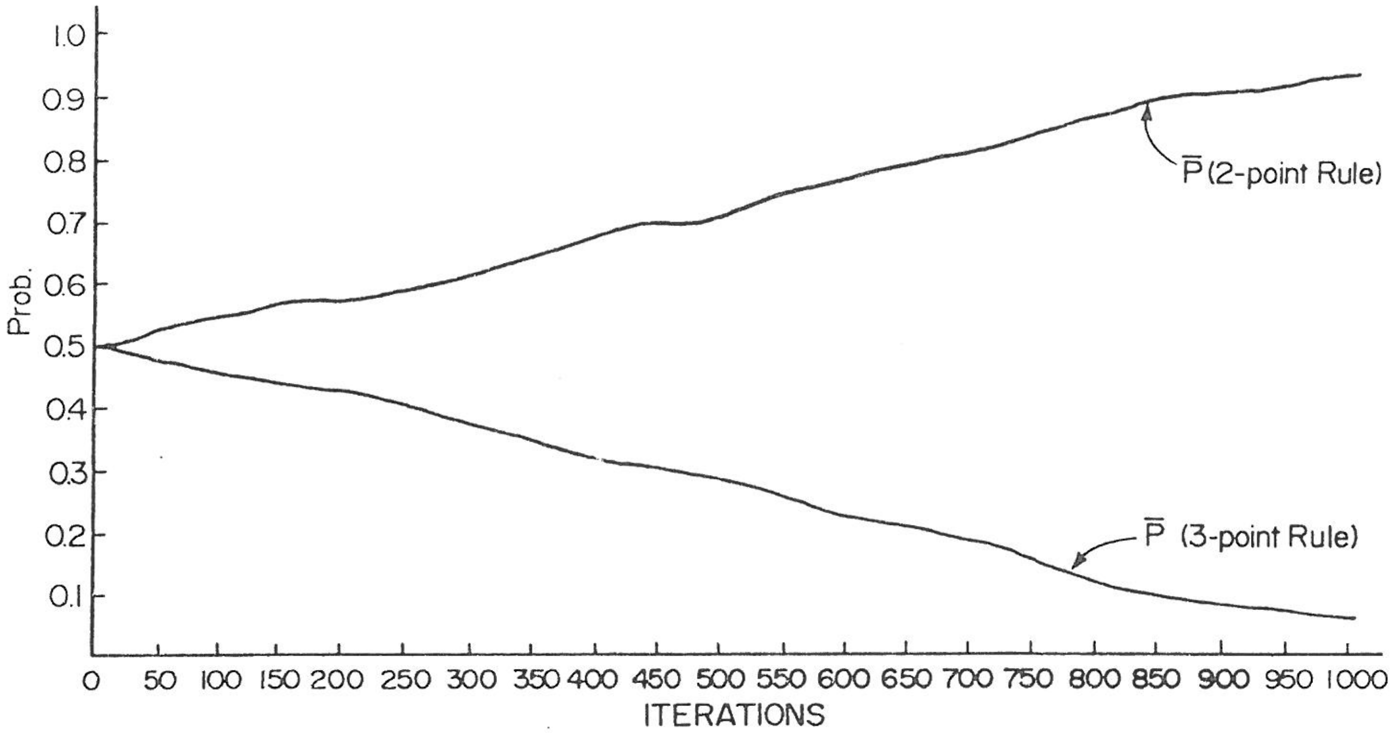

FIG. 11(b): Variation of the expected probabilities of choosing the 2-point and the 3-point rules for the data set shown in Fig. 11(a).

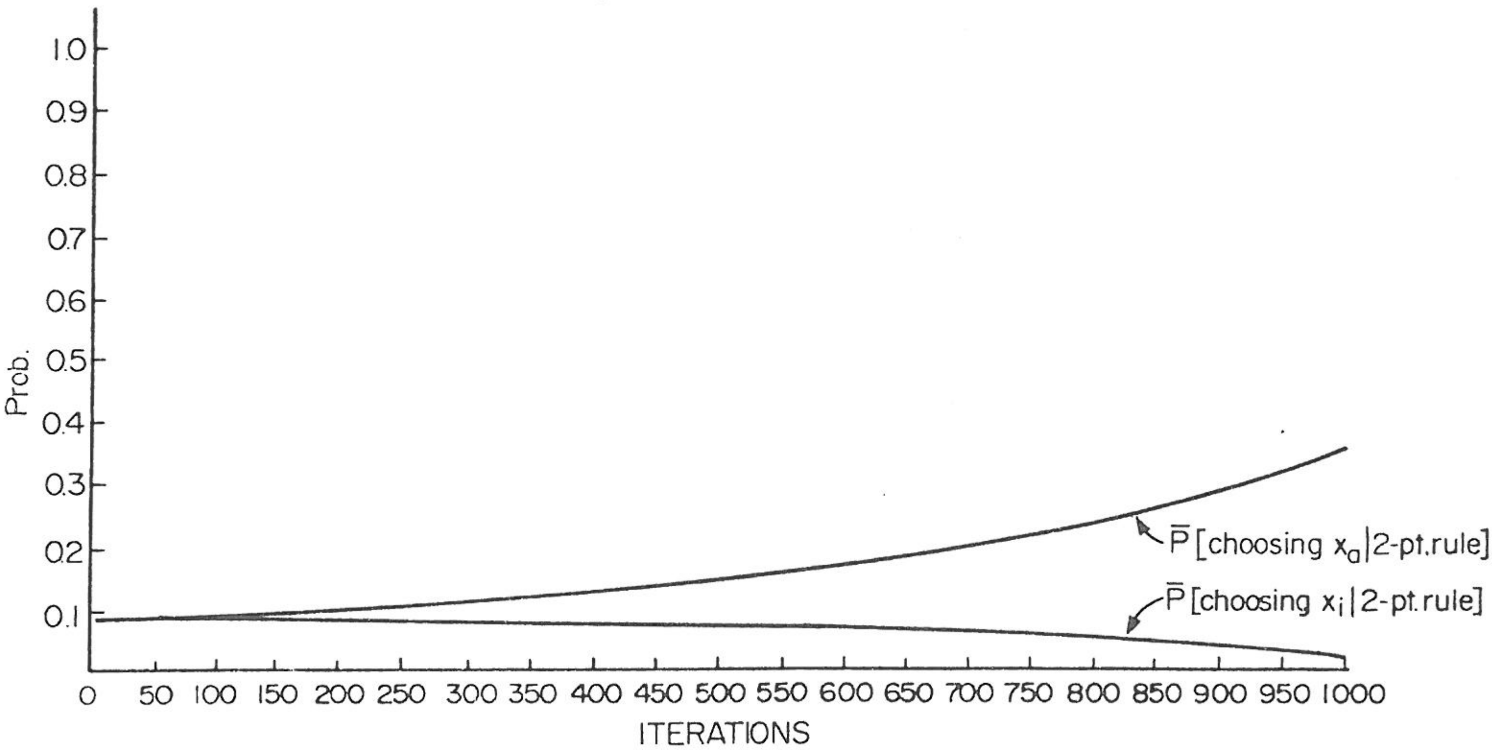

FIG. 11(c): Variation of the expected conditional probability of choosing x_a and any other x_i ($i \neq a,b$) for the data set of Fig. 11(a).

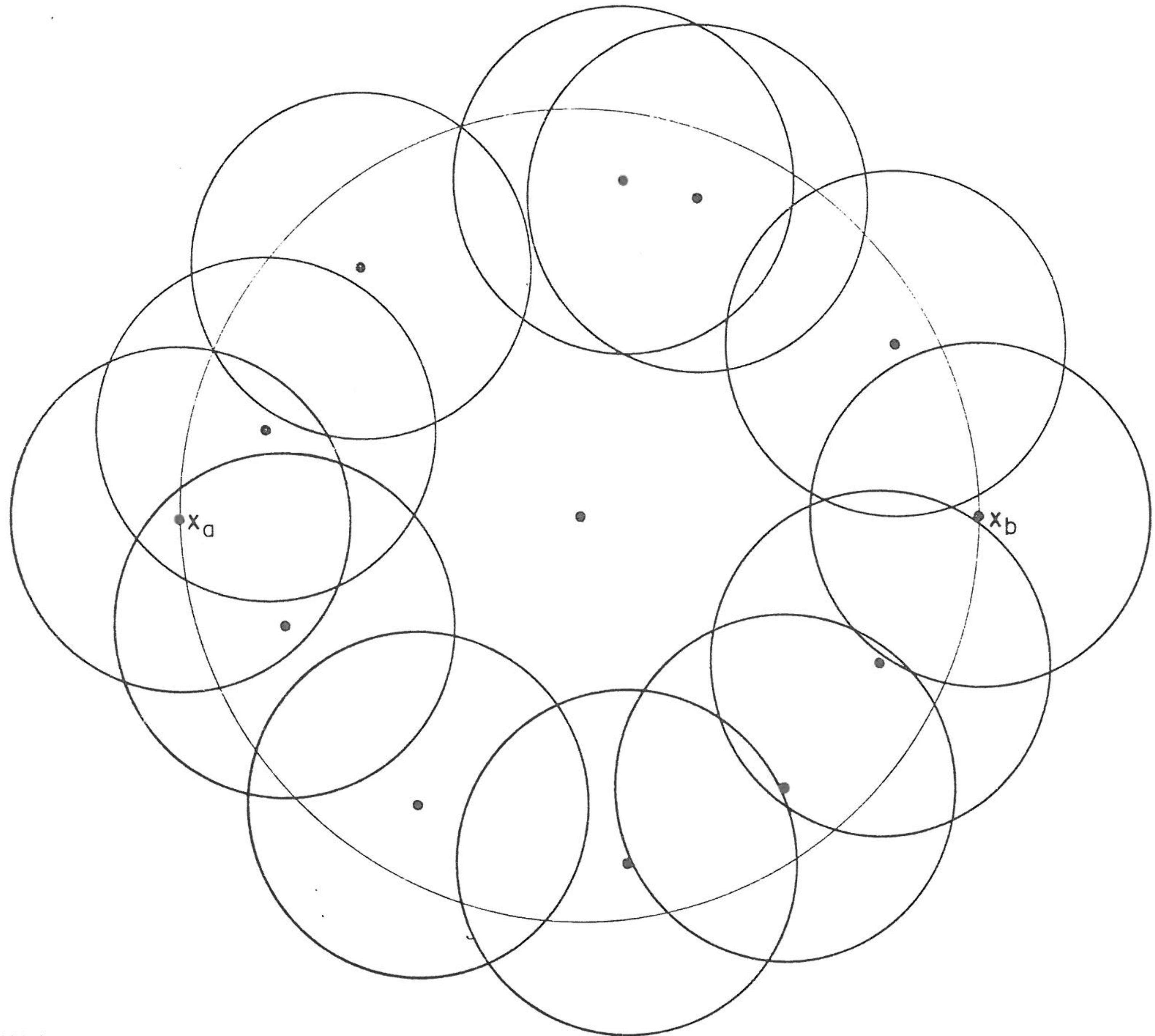

FIG. III(a): The set of points whose MSC is described by x_a, x_b. The small circles are 3 standard

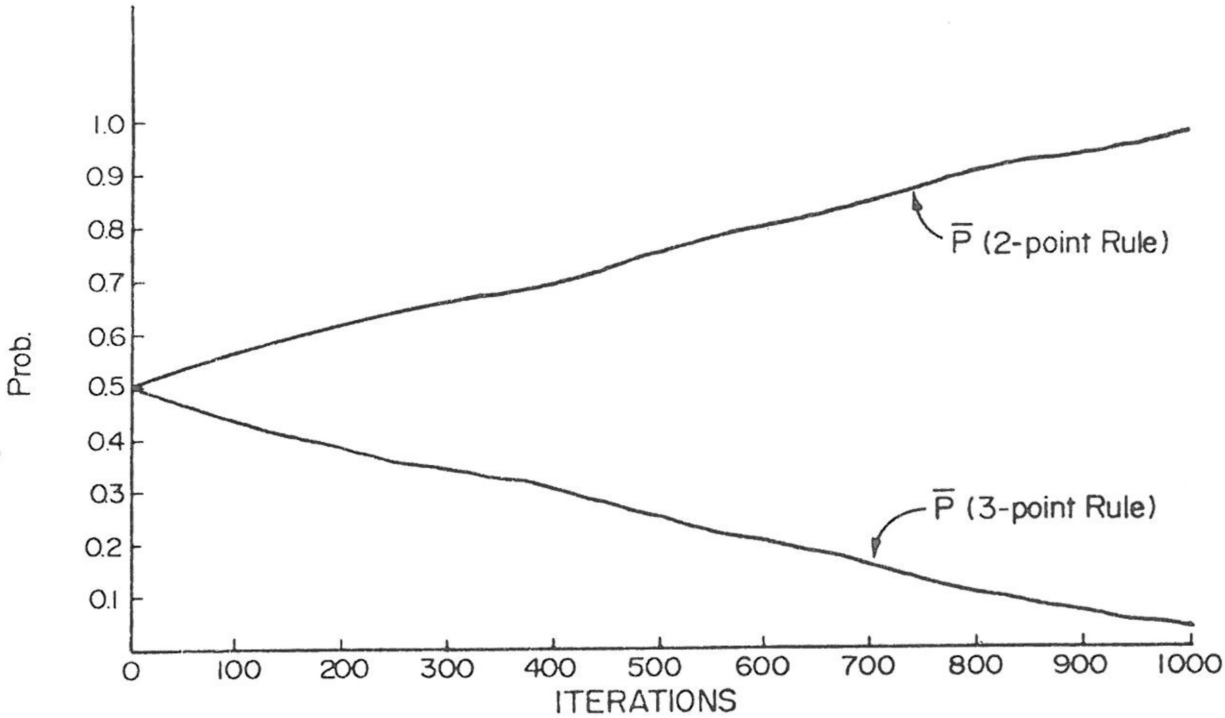

FIG. III(b): Variation of the expected probabilities of choosing the 2-point and 3-point rules for the data set shown in Figure III(a).

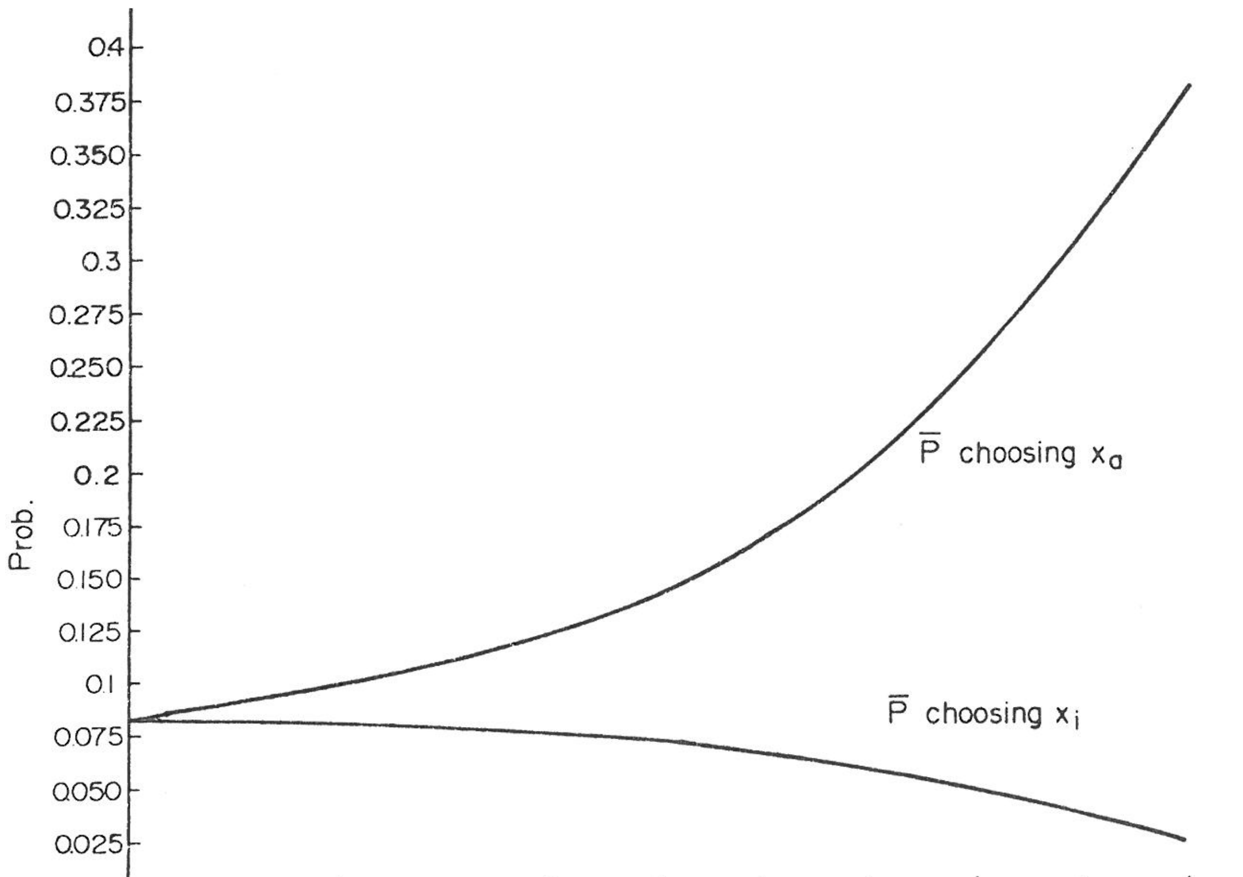

FIG. III(c): Variation of the expected probability of choosing x_a and any other x_i, $(i \neq a,b)$ for the data set of Figure III(a).

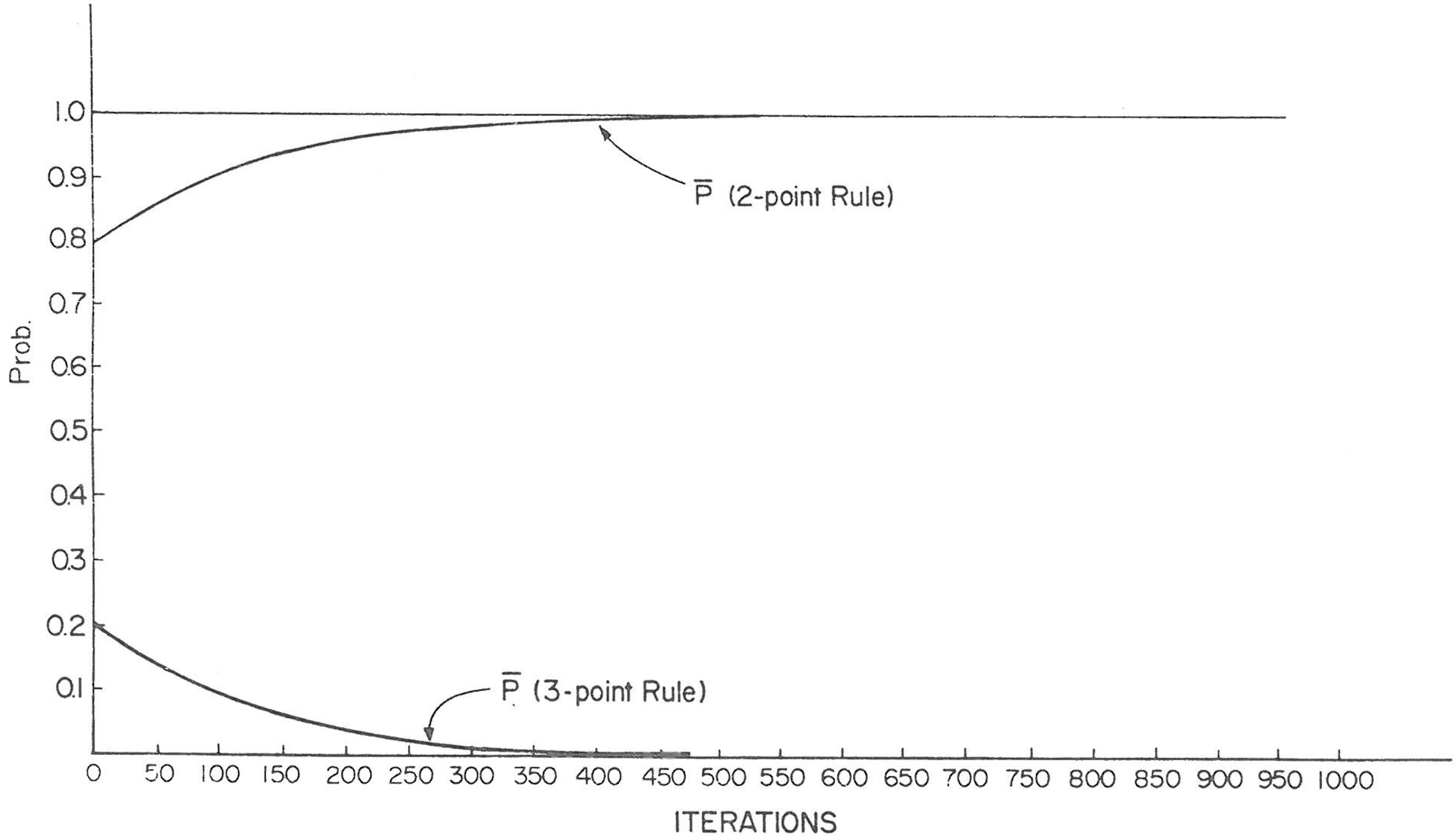

FIG. IV: Variation of the expected probability of choosing the 2-point and 3-point rules. The data set is as in Figure II(a). The initial probabilities are assigned values (0.8, 0.2) based on apriori information, thus hastening the convergence.