数理解析研究所講究録
第128巻 1971年 186-204

不確定オートマトンによる学習制御

北嶋靖三　　浅居喜代治
（大阪市立大学　工学部）（大阪府立大学　工学部）

§1. まえがき

制御対象の特性が未知でかつ多峰性の目的関数をもつ制御系の最適点を探索する方法としてはいろいろあるが，その一つの方法として目的関数の極値を学習的に求める学習制御がある。この学習制御系において，学習動作を行なわせるためには，内部状態の遷移が不確定な不確定オートマトンをコントローラとして用い，その遷移特性を学習によって変化させればよい。このようにすればオートマトンの内部で一種の自己組織動作が行なわれ，最適点を探索するのに最初は大局的探索が，次いで最適点近傍における局部探索が行なわれて早く最適点を探索することが可能となる。

この不確定オートマトンとしては，状態の遷移が遷移確率によって行なわれる確率オートマトン[1),(2)]とメンバシップ関数の概念[(3)]によって状態遷移が行なわれるあいまいオートマトン[(4)]

とが考えられる。ここではこれら二つのオートマトンの学習特性などについて比較検討した結果について述べる。

§2. 確率オートマトンによる学習制御

確率オートマトンはある状態から他の状態への遷移が遷移確率に基づいて行なわれるオートマトンで，次のような二つの形のものが考えられる。その一つは状態遷移が行なわれる際に出力が出される枝出力形のオートマトンで，他の一つは遷移が行なわれたのちにその状態に用意されているいくつかの出力の中から出力選択の確率に基づいて一つの出力が出される状態多出力形オートマトンである。これらの確率オートマトンの特性は次式で示される。

(a) 枝出力形確率オートマトン

$$M = \{ S, X, U, \{P(u/x)\} \} \qquad (1)$$

ただし，$S = \{s_1, s_2, \cdots, s_\nu\}$ ：ν個の状態の空でない集合，
$X = \{x_1, x_2, \cdots, x_\mu\}$：$\mu$個の入力の集合，$U = \{u_1, u_2 \cdots, u_\varsigma\}$：$\varsigma$個の出力の集合，$P(u/x)$：$\nu \times \nu$の確率遷移マトリクス。

一入力の場合の確率遷移マトリクスは

$$\left.\begin{aligned} P(u_k/x_1) &= [p_{ij}(u_k/x_1)] \\ \sum_{k=1}^{\varsigma} \sum_{j=1}^{\nu} p_{ij}(u_k/x_1) &= 1 \end{aligned}\right\} \qquad (2)$$

ただし，$u_k \in U$，$x_1 \in X$，$k=1,2,\cdots,\xi$，$i,j=1,2,\cdots,\nu$，$p_{ij}(u_k/x_1)$：入力x_1が加えられたとき，状態s_iから状態s_jへ遷移が行なわれ，かつ出力u_kを出す確率，$p_{ij}(u_k/x_1) \geq 0$。

(b) 状態多出力形確率オートマトン

$$M = \{S, X, U, \{P(x)\}, Q(s)\} \qquad (3)$$

ただし，S, X, U：(1)式の場合と同様，$P(x)$：$\nu \times \nu$の確率遷移マトリクス，$Q(s)$：出力選択の確率マトリクス($\nu \times \xi$)。一入力の場合を考えると，

$$\left.\begin{aligned} P(x_1) &= [p_{ij}(x_1)] \quad , \quad \sum_{j=1}^{\nu} p_{ij}(x_1) = 1 \\ Q(s) &= [g_{jk}] \qquad\quad \sum_{k=1}^{\xi} g_{jk} = 1 \end{aligned}\right\} \qquad (4)$$

これらの確率オートマトンを用いた場合の学習制御系を図1に示す。

図において，制御対象は確率オートマトンの出力に対応した操作量uが加えられると，出力yを出す。目的関数評価部はuおよびyとから系の目的関数を計算し，この目的関数の値が最適値に近づいているか(成功という)あるいは遠ざかっているか(失敗という)を判定し，成功信号あるいは失敗信号を遷移確率修正部へ送る。遷移確率修正部では，これらの信号が加えられると，成功信号の場合は確率オートマトンの遷移確率が大きく，失敗信号の場合は遷移確率が小さくなるように，遷移が行なわれた枝の遷移確率を修正する。確率オー

トマトンに次の入力が加えられると，新しく修正された確率遷移マトリクスに基づいて遷移が行なわれる。

このような動作が繰り返し行なわれて，制御対象への最適操作量が決定される。また，確率オートマトンの内部では最適操作量が選ばれるような枝または状態を選択する確率が大きくなり，一種の自己組織動作が行なわれたことになる。

これらのオートマトンにおいて，確率の修正は次式で示される linear reinforcement algorithm によって行なわれる。

$$p_{ij}(n+1) = \alpha p_{ij}(n) + (1-\alpha)\lambda_{ij}(n) \qquad (5)$$

$$p_{ik}(n+1) = \alpha p_{ik}(n) + (1-\alpha)\frac{1-\lambda_{ij}(n)}{\nu-1} \qquad (6)$$
$$(k \neq j)$$

ただし，

$$\lambda_{ij}(n) = \begin{cases} \dfrac{I(n)}{I_{max}} & \text{成功のとき } (I \geq \bar{I}) \\ 0 & \text{失敗のとき } (I < \bar{I}) \end{cases},$$

$I(n)$：目的関数の値，I_{max}：現在までの I の最大値

この場合 $p_{ij}(n+1)$ 以外の確率はすべて一様に $(1-\lambda_{ij}(n))/\nu-1$ に収れんするように修正される。したがって学習の初期においては修正量の変動が大きくなる。この変動を小さくするためには，修正時における確率分布に従って次式のように荷重 $\overline{w}$ を付加すればよい。

$$p_{ik}(n+1) = \alpha p_{ik}(n) + (1-\alpha)(1-\lambda_{ij}(n))\overline{w} \qquad (7)$$
$$(k \neq j)$$

ただし,

$$W = \frac{p_{ik}(n)}{1 - p_{ij}(n)}$$

また,

$$\overline{W} = \frac{\frac{p_{ij}(n) - p_{ij}(n+1)}{(\nu-1)(1-\alpha)} + p_{ik}(n)}{1 - \lambda_{ij}(n)} \qquad (8)$$

とし,

$$p_{ik}(n+1) = \alpha\, p_{ik}(n) + (1-\alpha)\left\{p_{ik}(n) + \frac{p_{ij}(n) - p_{ij}(n+1)}{(\nu-1)(1-\alpha)}\right\}$$

$$= p_{ik}(n) + \frac{p_{ij}(n) - p_{ij}(n+1)}{\nu - 1} \qquad (9)$$

のように $p_{ik}(n+1)$ の收れん値を $p_{ij}(n) - p_{ij}(n+1)$ によって修正する方法も考えられる。

この linear reinforcement algorithm に用いられている係数 α は收れんの速さに関するもので，この値が小さい程收れんが速くなる。したがって，この α の値を系の目的関数の値に対応して変化させるようにすれば，学習の初期においては α の値が小さく，学習の進行に従って大きくすることができ，最適点を速く探索することができるとともに，ハンチングロスの少ない制御を行なわせることができる。

この考えに基づいて，ここでは α の値を次のように定めている。

$$\alpha = 1 - \left|\frac{I(n) - \overline{I}(n)}{\overline{I}(n)}\right| \qquad (10)$$

ただし，　$\bar{I}(n)$：目的関数の累積平均値

$I(n)$が $0 < I(n) < \bar{I}(n)$ あるいは $\bar{I}(n) < I(n) < 2\bar{I}(n)$ の範囲にあれば，αの値は $0 < \alpha < 1$ となり，$f_{ij}(n+1)$ は $\lambda_{ij}(n)$ に收れんすることとなる。また，$I(n) = \bar{I}(n)$ の場合には $\alpha = 1$ となり $f_{ij}(n+1)$ の値は修正されない。

以上述べた学習制御系において，枝出力形確率オートマトンを用いる場合には，枝の遷移確率を上記の方法で修正することにより最適点の探索を行なわせることができる。

また状態多出力形を用いる場合には，遷移が行なわれたのちの状態から一つの出力が選択されることから，ある出力が選び出される確率 g_k は，

$$g_k = [\pi_i][p_{ij}][g_{jk}][\eta_k] \qquad (11)$$

ただし，　$i, j = 1, 2, \cdots, \nu$ ，$k = 1, 2, \cdots, s$ ，

$[\pi_i]$：状態分布行ベクトル，$[p_{ij}]$：確率遷移マトリクス，

$[g_{jk}]$：出力選択マトリクス，$[\eta_k]$：k番目が1の列ベクトル $\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$，

となり，試行の成功あるいは失敗によって p_{ij} および g_{jk} の二つの確率が上記の方法で修正される。

この場合の学習特性は，(i) 遷移確率 p_{ij} によってある状態が選ばれ，成功か失敗かによって p_{ij} が (5), (9) および (10) 式を用いて修正される（大局的探索）。(ii) ある状態が選ばれると，その状態内での出力が出力選択確率 g_{jk} によって選

ばれる。この場合，この状態における出力に対応した目的関数の平均値を求めるために何回かの探索が行なわれる（局部的探索）。

このように(i)と(ii)とが交互に繰り返されて最適点が探索される。

§3. あいまいオートマトンによる学習制御

あいまいオートマトンは筆者らによって提案されたものであり，状態間の遷移がメンバシップ関数（以下MFと略記）に基づいて行なわれるものである。この場合も確率オートマトンと同様枝出力形および状態多出力形のものが考えられる。

枝出力形および状態出力形あいまいオートマトンは次のように表わされる。

(a) 枝出力形あいまいオートマトン

$$M = \{S, X, U, \{F(u/x)\}\} \tag{12}$$

ただし，S, X, U：(1)式の場合と同じ，$F(u/x)$：$\nu \times \nu$ あいまい遷移マトリクス。

一入力の場合のあいまい遷移マトリクスは，

$$F(u_k/x_1) = [f_{ij}(u_k/x_1)] \tag{13}$$

ただし，$u_k \in U$，$x_1 \in X$，$k = 1, 2, \cdots, \xi$，$i, j = 1, 2, \cdots, \nu$，$f_{ij}(u_k/x_1)$：入力x_1がきて出力u_kを出す場合の，状態S_i

から状態 s_j への枝による遷移に対するMF，$0 \leq f_{ij}(u_k/x_i) \leq 1$。

(b)状態多出力形あいまいオートマトン

$$M = \{S, X, U, \{F(x)\}, G(s,u)\} \qquad (14)$$

ただし，$F(x)$：$V \times V$ あいまい遷移マトリクス，$G(s,u)$：$V \times S$ の出力選択マトリクス。

これらのあいまいオートマトンを用いた学習制御系を図2に示す。

MFの修正法は確率オートマトンの場合と同様 linear reinforcement algorithm を用いているが，確率オートマトンとは異なり，遷移が行なわれた枝のMFのみが修正される。

$$f_{ij}(n+1) = \alpha f_{ij}(n) + (1-\alpha)\lambda_{ij}(n) \qquad (15)$$

ただし，
$$\lambda_{ij}(n) = \begin{cases} I(n)/I_{max} & \text{成功の場合} \\ 0 & \text{失敗の場合} \end{cases}$$

枝出力形あいまいオートマトンを用いた場合の学習特性は，高次遷移特性を用いた大局的探索に特徴があり，学習が進むに従って探索の範囲が縮小されて最適点が探索される。この場合オートマトン内部では最適探索に寄与した枝のMFは1に，そうでない枝のMFは0に近づくので，明らかに自己組織動作が行なわれていることがわかる。

状態多出力形の場合は，(15)式に示された f_{ij} の修正以外に出力選択のMFの修正が行なわれる。

$$g_{jk}(n+1) = \alpha\, g_{jk}(n) + (1-\alpha)\lambda_{jk}(n) \qquad (16)$$

この場合の学習特性は次のようになる。

(i) どの状態に最適点が存在するかを推定するために高次遷移特性を利用して各状態間を探索する(大局的探索)。この場合図2においてスイッチ S_1 および S_2 は端子"1"へ接続される。(ii) ある状態に最適点が存在すると推定されたならば,その状態におけるいくつかの出力を探索する(局部的探索)。スイッチ S_1 および S_2 は端子"2"へ接続される。(iii) ある状態に最適点が存在する可能性が強くなれば,その状態をとり易くする(自己ループの形成)。スイッチ S_1 および S_2 は端子"3"に接続される。(iv) ある状態に最適点が存在し得ないと判定されれば,その状態をとり難くする(淘汰)。スイッチ S_1 および S_2 は"4"に接続される。

§4. シミュレーション実験

以上述べた確率オートマトンおよびあいまいオートマトンを用いた場合の学習制御系のシミュレーション実験を,目的関数の応答面が双峰性のものについて行なった。その結果を図3～図7に示す。

図から明らかなように,確率オートマトンおよびあいまいオートマトンのどちらの場合も局部的最適点を避けて真の最

適点に到達しており，その学習特性は類似している。また状態多出力形オートマトンを用いた場合の方が枝出力形の場合に比べて最適点への到達が速い。あいまいオートマトンと確率オートマトンとを用いた場合を比較すると，あいまいオートマトンの方は最適点へ到達するためのパスが不確定なものから次第に確定的なものとなるため速く最適点に到達し，最適点到達後はその点を保持するためのパスが決定されるためにハンチングロスが少なくなる。これに対し確率オートマトンの場合は，確率に基づいて遷移が行なわれるためランダム性があり，また遷移が行なわれた枝以外の枝の確率も同時に修正されるため最適点への到達が遅くなる。最適点へ到達したのちも，その点以外の点をとる場合があり，ハンチングロスが大きくなる。

オートマトンの構造が学習によって変化する様子を検討するために，遷移確率 p_{ij}，g_{jk} および遷移MF f_{ij}，g_{jk} の変化を探索回数を横軸にとって示したのが図8である。いずれの場合も linear reinforcement algorithm によって確率またはMFの修正が行なわれるために収れんの様子は類似しているが，その速さはMFの方が速く1に収れんしている。

§5 むすび

以上確率オートマトンおよびあいまいオートマトンの2種の不確定オートマトンを学習制御系のコントローラとして用いた場合の学習特性について述べたが，どちらの場合も学習の進行に従って自己組織動作を行ない，局部的最適点を避けて，真の最適点に到達している。しかし，確率オートマトンの場合は，確率に基づいて遷移が行なわれるためのランダム性があり，また遷移が行なわれた枝以外の枝の確率も同時に修正されるため最適点への到達が遅い。あいまいオートマトンの場合は，学習が進むに従って最適点を選ぶパスが確定的になるため最適点への到達が速く，オートマトン内部における自己組織動作が非常に明らかに現われている。また，演算操作は比較演算のみであるから簡単である。

§6 参考文献

(1) G.M. McMurtry and K.S.Fu : A variable structure automaton used as a multimodal searching technique, IEEE. Trans. on A.C., AC-11, 3, pp. 379~387, July (1966).

(2) J.M. Mendel and K.S. Fu : Adaptive learning, and pattern recognition systems, Academic Press, 1970, pp. 393~431

(3) L.A. Zadeh : Fuzzy sets, Information and control, 8, 3,

pp. 338～353, June (1965).

(4) 北嶋・浅居："あいまいオートマトンによる学習制御", 14, 9, pp. 551～559, 9月(昭和45年).

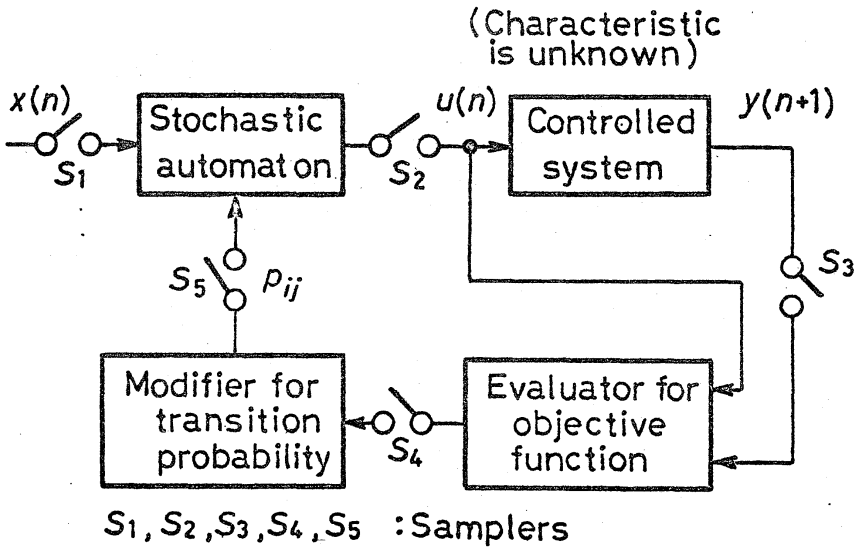
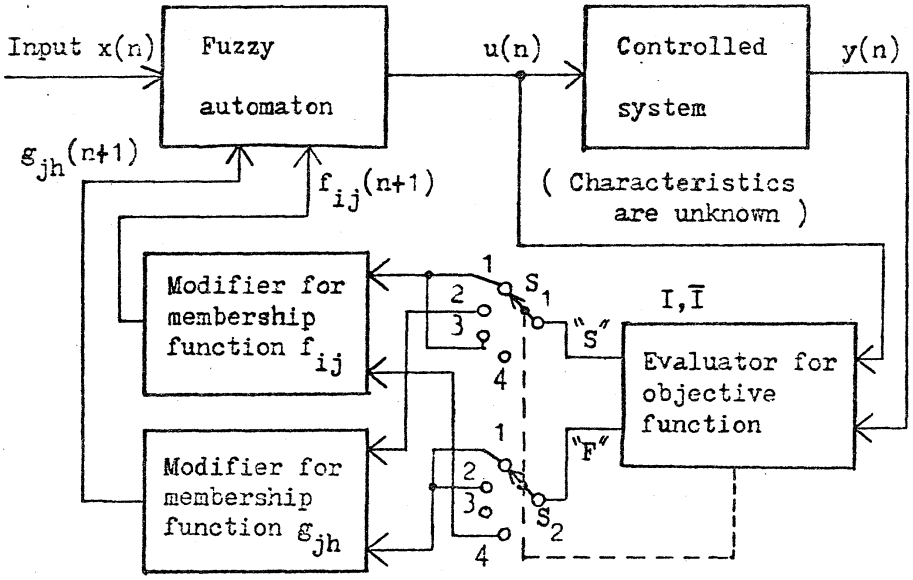
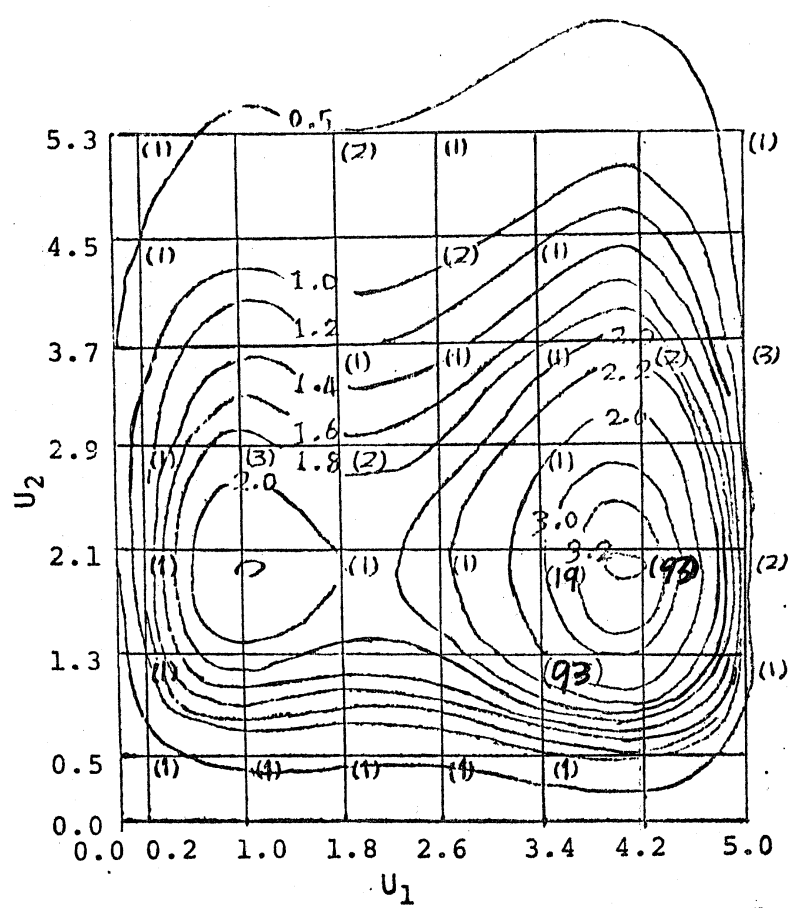


Fig. 1 Block diagram of learning control system using stochastic automaton.



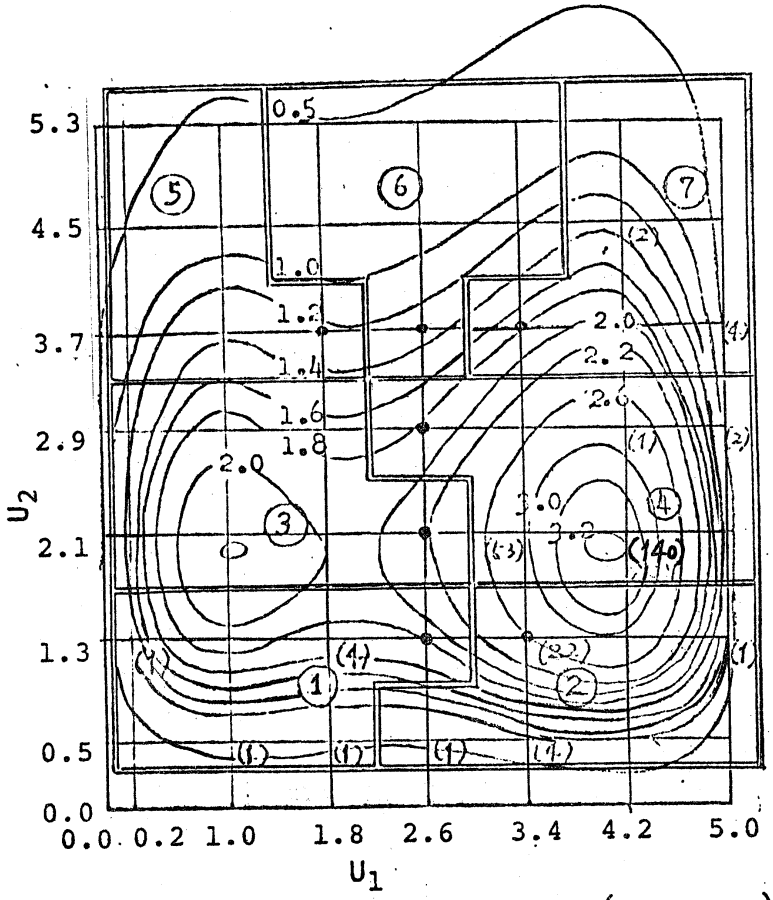
"S" : Success signal , "F" : Failure signal , S_1 and S_2 : Switches , 1 : Global search , 2 : Local search , 3 : Self loop , 4 : Selection .

Fig. 2 Block diagram of learning control system using fuzzy automaton.



●: TRIAL POINTS AT STARTING TIME(7 POINTS),
×: TRIAL POINTS AT FINAL STAGE(2 POINTS),
⊙: OPTIMUM POINT, ⊡: LOCAL OPTIMUM POINT,
△: SADDLE POINT, ( ): NUMBER OF TRIALS.

(a) Mealy type of stochastic automaton.



●: TRIAL POINTS AT STARTING TIME(7 POINTS),
×: TRIAL POINTS AT FINAL STAGE(1 POINT),
⊙: OPTIMUM POINT, ⊡: LOCAL OPTIMUM POINT,
△: SADDLE POINT, ( ): NUMBER OF TRIALS,
○: STATE NUMBER.

(b) Moore type of stochastic automaton.

Fig. 3 Number of trials on responce surface of objective function.

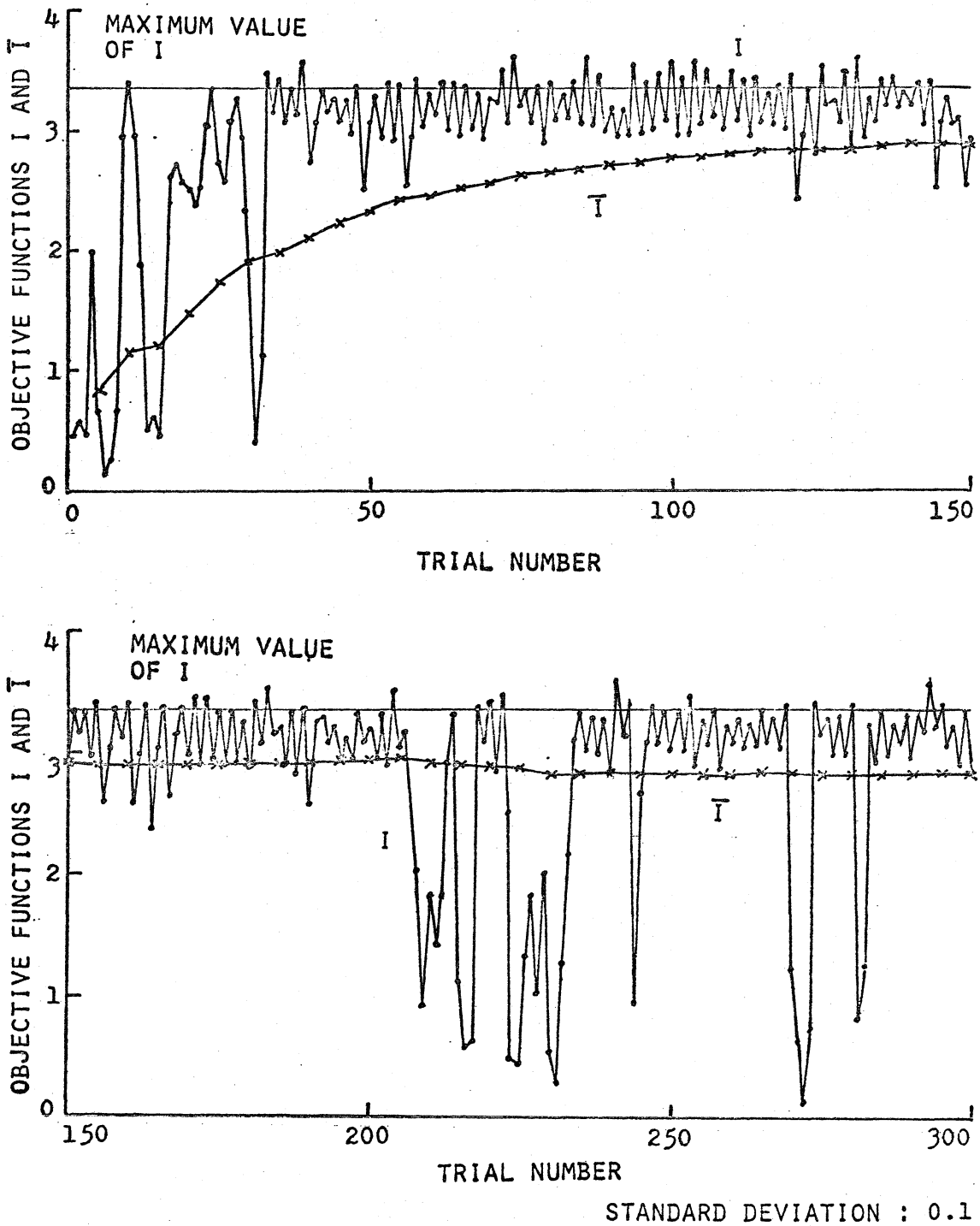


Fig. 4 An example of results of simulation study (case of Mealy type of stochastic automaton).

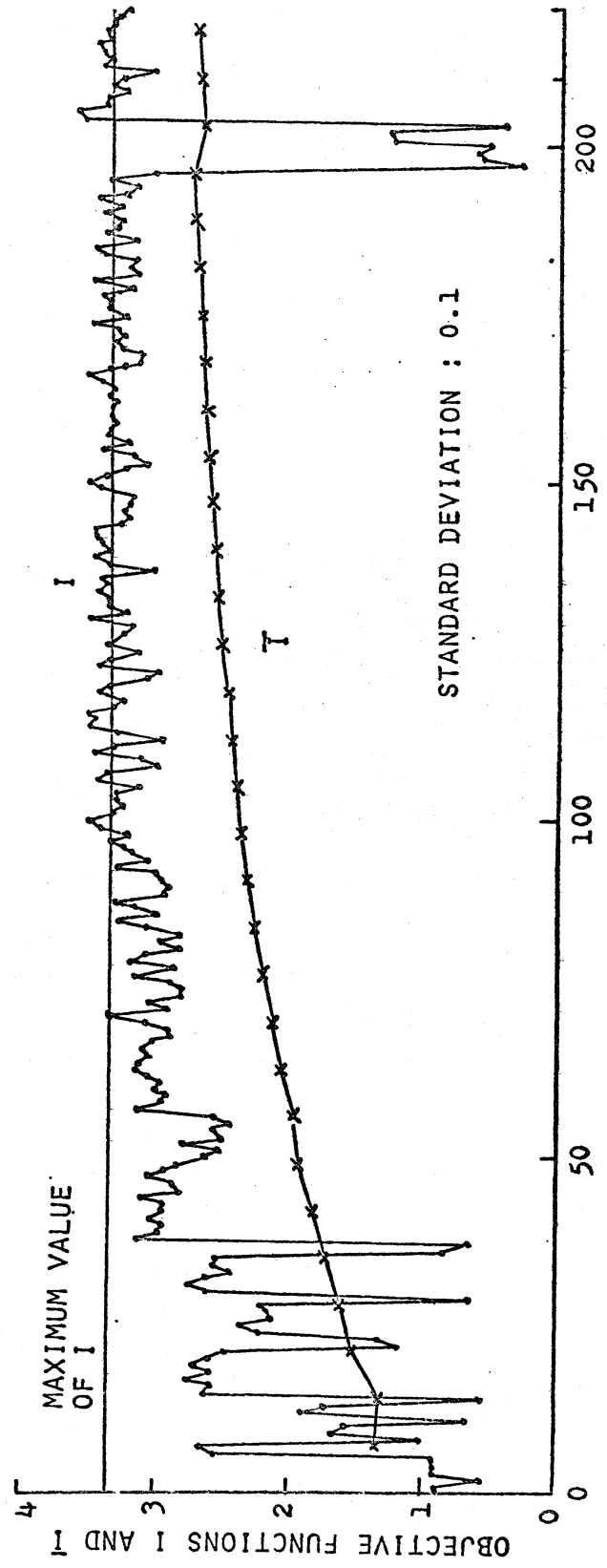
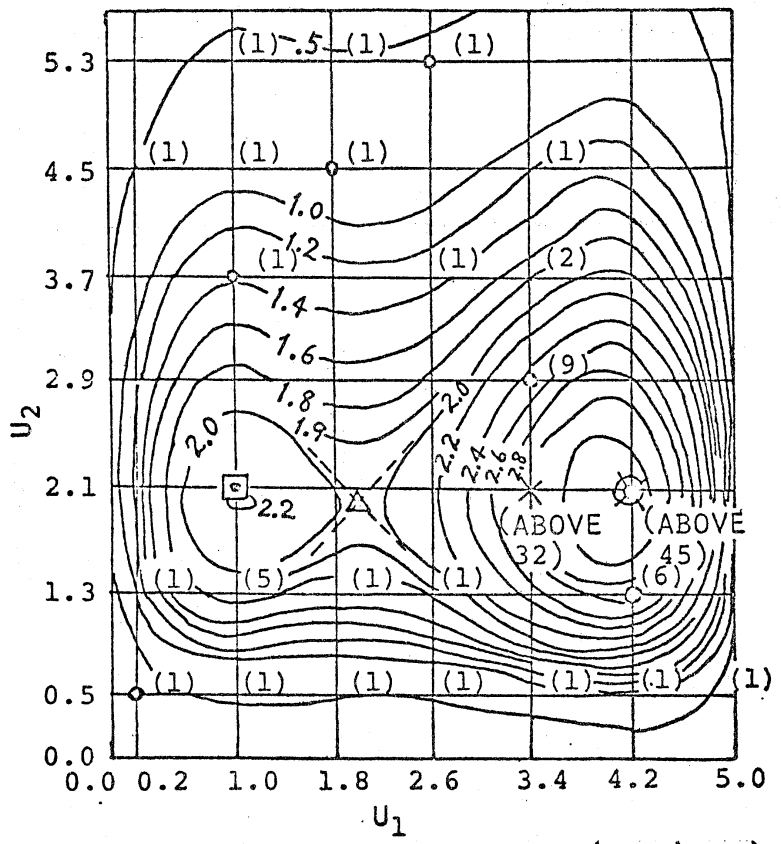


Fig. 5 An example of results of simulation study (case of Moore type of stochastic automaton).



●: TRIAL POINTS AT STARTING TIME(7 POINTS),
×: TRIAL POINTS AT FINAL STAGE(2 POINTS),
⊙: OPTIMUM POINT, ⊡: LOCAL OPTIMUM POINT,
△: SADDLE POINT, ( ): NUMBER OF TRIALS.

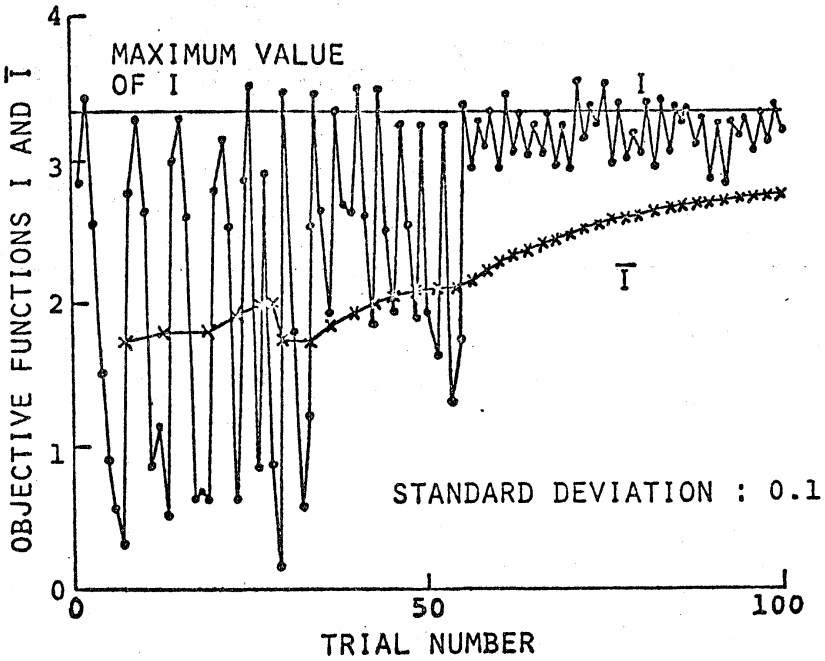
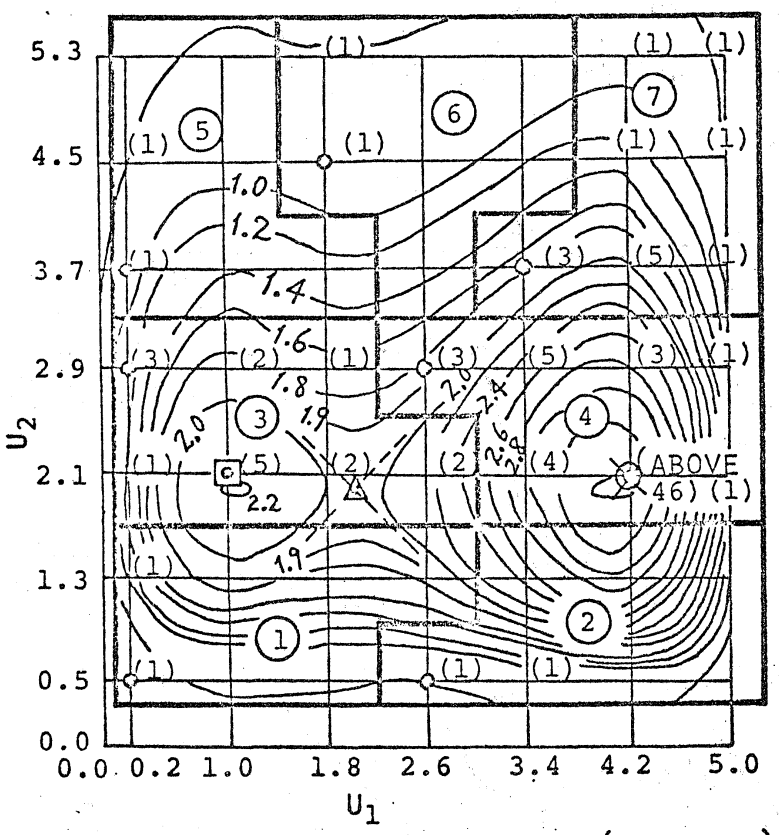


Fig. 6

An example of results of simulation study (case of Mealy type of fuzzy automaton).



o: TRIAL POINTS AT STARTING TIME(7 POINTS),
×: TRIAL POINTS AT FINAL STAGE(1 POINT),
◎: OPTIMUM POINT, ⊡: LOCAL OPTIMUM POINT,
△: SADDLE POINT, ( ): NUMBER OF TRIALS,
○: STATE NUMBER.

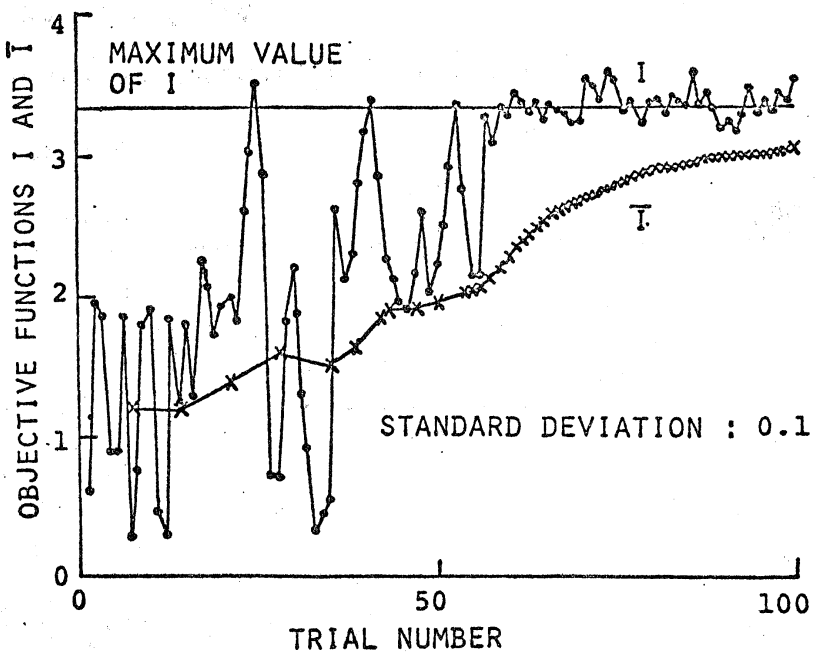


Fig. 7

An example of results of simulation study (case of Moore type of fuzzy automaton).

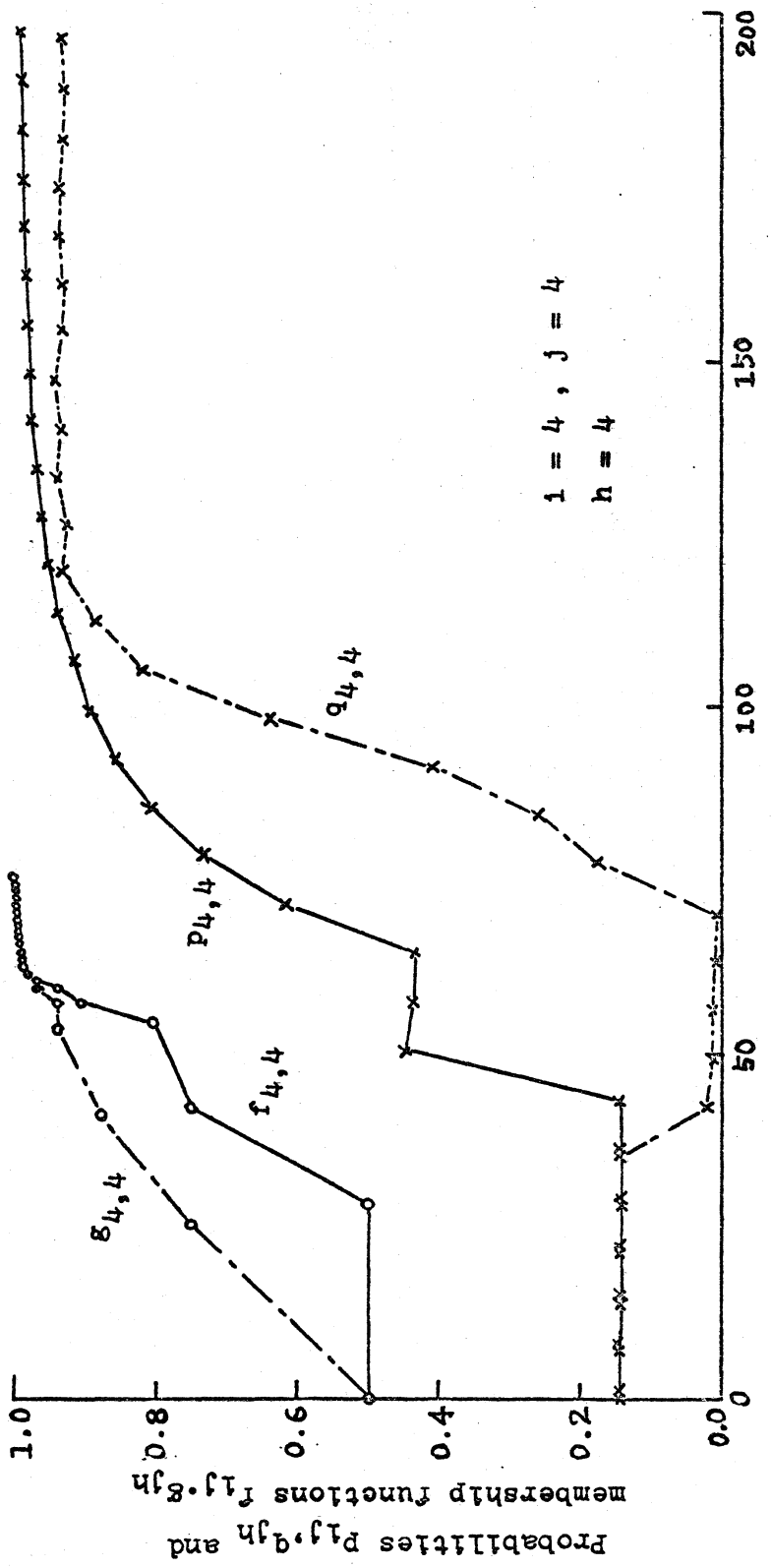


Fig. 8 Convergent behaviours of probabilities p_{ij}, q_{jh} and membership functions f_{ij}, g_{jh}.