

あいまいオートマトンと

その学習制御への応用

平井平八郎 浅居喜代治 北嶋靖三

(大阪市立大学工学部)

1. まえがき

従来の集合論においては、ものの集りが集合であるためには、その集められるものの範囲がはっきりと定まっていなくてはならなかったが、この範囲が不明確な場合にも集合論を拡張しようとする試みが L.A.Zadeh によってなされ、その考え方に対してあいまい集合論 (Concepts of Fuzzy Set) という名称が付けられた。⁽¹⁾⁽²⁾ その中心になる思想はいくつかの要素がある集合に含まれる度合い (Grade) を示すのにメンバーシップ関数 (Membership Functions, 以下 MF と略記) を考えるもので、その数学的取扱いに対して種々の定義が与えられている。この MF の数値は $[0, 1]$ の範囲をとるものと定められているが、 $MF = 1$ の場合にある集合はその要素を完全に含み、また $MF = 0$ の場合にある集合はその要素を完全に含まないということになり、これが従来の

集合論の場合に相当する。

筆者らは、この MF に基づいて遷移の行なわれるオートマトンとして二つの型のを構成し（以下あいまいオートマトンという）、その学習特性を検討して学習制御に用いるのに有効であることを明らかにしたので以下に報告する。

2. あいまいオートマトン

あいまいオートマトンとは、入力がくるごとに、ある状態から他の状態またはもとの状態へ MF が最大の枝（Branch）を選んで遷移を行なうオートマトンである。これには出力を、遷移した枝または遷移後の状態のいずれに対応して出すかによって二つの型のものが考えられる。ここでは便宜上、前者を枝出力形、後者を状態出力形と呼ぶことにする。

あいまいオートマトン M を次のように記述する。

$$M = \{ S, X, F, U \} \quad \text{---- (1)}$$

ここで $S = \{ s_1, s_2, \dots, s_\nu \}$: ν 個の状態集合

$X = \{ x_1, x_2, \dots, x_\mu \}$: μ 個の入力の集合

F : MF によって構成せられる遷移マトリクス

$U : \{ u_1, u_2, \dots, u_\xi \}$: ξ 個の出力の集合

F および U を二つの型について具体的に示すと次のようになる。

(a) 枝出力形

$$F = [f_{ij}(u/x)] = \begin{pmatrix} f_{11}(u/x) & f_{12}(u/x) & \cdots & f_{1v}(u/x) \\ f_{21}(u/x) & f_{22}(u/x) & \cdots & f_{2v}(u/x) \\ \vdots & \vdots & \ddots & \vdots \\ f_{iv}(u/x) & & & \\ \vdots & & & \\ f_{v1}(u/x) & f_{v2}(u/x) & \cdots & f_{vv}(u/x) \end{pmatrix} \quad \cdots (2)$$

ここで $f_{ij}(u/x)$: 入力 x がきて出力 u 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, $0 \leq f_{ij} \leq 1$

$$U = \{u_1, u_2, \dots, u_s\} \quad \cdots (3)$$

(b) 状態出力形

$$F = [f_{ij}(x)] = \begin{pmatrix} f_{11}(x) & f_{12}(x) & \cdots & f_{1v}(x) \\ f_{21}(x) & f_{22}(x) & \cdots & f_{2v}(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_{iv}(x) & & & \\ \vdots & & & \\ f_{v1}(x) & f_{v2}(x) & \cdots & f_{vv}(x) \end{pmatrix} \quad \cdots (4)$$

ここで $f_{ij}(x)$: 入力 x がくる場合の, 状態 S_i から状態
 S_j への枝による遷移に対する MF,
 $0 \leq f_{ij} \leq 1$

$$U = \varphi(S) \quad \text{----- (5)}$$

$$\text{ここで } U = \{u_1, u_2, \dots, u_v\}$$

以上二つの型の相異点を要約すると次のとおりである。

(i) 遷移マトリクス F は、枝出力形の場合には入力 x と出力 u との組合わせ数 u_x だけ、また状態出力形の場合には入力 x の数 u だけそれぞれ必要である。

(ii) 出力集合 U は状態出力形の場合には v 個の要素をもつ。

以上いおれの型においても、ある状態から二つ以上の直列の枝を経て遷移する高次遷移 (Higher-order Transition) の場合の遷移マトリクスはあいまい集合論の定義⁽¹⁾から次のように示すことができる。

$$F^{(k)} = [f_{ij}^{(k)}] \quad \text{----- (6)}$$

$$\begin{aligned} f_{ij}^{(k)} &= \{S_i, x^{(k)}, S_j\} \\ &= \text{Sup } S_p, S_q, \dots, S_v \text{ Min } [f_{ip}\{S_i, x^{(1)}, S_p\}, f_{pq}\{S_p, x^{(2)}, S_q\}, \\ &\quad \dots, f_{vj}\{S_v, x^{(k)}, S_j\}] \dots (7) \end{aligned}$$

ここで、 k : 直列の枝の数

$S_p, S_q, \dots, S_v$: 状態 S_i から状態 S_j に至るパス^{*}の途中にある状態

式(6), (7)によって示される遷移マトリクスを高次遷移マト

^{*} 二つ以上の枝から成る直列のものをパス (Path) という。

リクスと呼びが、これによってある状態からの各遷移のパ
スが決定論的 (deterministic) に定められる。

以上のようなあいまいオートマトンは、特性未知の制御対
象の最適制御またはパターン認識に用いることができる。い
ずれの場合にも制御対象またはパターンからサンプリングさ
れた情報に基づき学習的に遷移マトリクスの要素である MF
を修正して最適点または正しいパターンの探求を行なう。こ
こでは制御の場合について以上二つの型のあいまいオートマ
トンの応用例を示す。

3. 学習制御への応用^{(3)~(5)}

あいまいオートマトンを用いる学習制御系の構成図を図 1
に示す。ここで制御対象の特性が未知で、かつその特性検知
のための制御対象入出力値に測定誤差が含まれ、さらにその
特性が外乱によって変動するときに、目的関数を最大にする
場合を取り扱う。

この系の動作を枝出力形と状態出力形とに分けて次に説明
する。

(a) 枝出力形

ここでは、簡単のためにあいまいオートマトンとして次式
で示される、一つの枝に一つの出力を対応させたものを用い

る。

$$X = \{x_1\} \quad \text{-----} \quad (8)$$

$$F = [f_{ij}(u_{ij})]_{x_1} = \begin{bmatrix} f_{11}(u_{11}) & f_{12}(u_{12}) & \cdots & f_{1v}(u_{1v}) \\ f_{21}(u_{21}) & f_{22}(u_{22}) & \cdots & f_{2v}(u_{2v}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{ij}(u_{ij}) & & & \\ \vdots & & & \\ f_{v1}(u_{v1}) & f_{v2}(u_{v2}) & \cdots & f_{vv}(u_{vv}) \end{bmatrix} \quad \text{---(9)}$$

$$U = \{u_{ij}\} \quad \text{-----} \quad (10)$$

ここで, u_{ij} : 入力 x_1 がくる場合に, 状態 S_i から状態 S_j に遷移が行なわれるときの出力

$i, j: 1, 2, \dots, v$

(i) あいまいオートマトンでは, 初めにすべての MF を 0 と 1 との中央値 0.5 に定めてあり, 入力 x_1 によって, ある状態から任意の状態へランダムに遷移が行なわれるようにしてある。

(ii) 次に, ある状態 S_i から同じ状態へ K 個の枝を経てループをなすパスを見つけるために, K 次遷移マトリクスが式(6)によって計算され, マトリクスの対角項にある MF のうちの最大値 $f_{ii}^{(K), \max}$ と, このような最大値をとる MF の個数 M とが求められる。

(iii) M を K と比較して、もしも $M < K$ ならば、重複する枝を取り除くために M 次遷移マトリクスが計算され、さらに (ii) と同様な手順が繰り返される。この手順は、すべての重複する枝がループ・パスから取り除かれるまで繰り返される。このようにして得られた MF の最大値および最大の MF の個数をそれぞれ $f_{i,j}^{(k), \max}$ および m とする。

(iv) S_i から S_j への単一のループ上の各個の枝はそれぞれ $f_{i,j}^{(k), \max}$ に等しいかまたは大きい MF をもつべきであるから、目的のループ・パスは $f_{i,j}^{(k), \max}$ を、 S_i から S_j への各個の枝の MF と比較することにより見出すことができる。ここでループ・パスが存在しないときは、 $(k-1)$ 次またはそれ以下の次数の遷移マトリクスについて以上の手順が繰り返される。

(v) (ii) ~ (iv) の手順によってループ・パスが求められると、このパスを経て遷移が行われ、各枝に対応する各個の出力が相次いでオートマトンから制御対象に送られる。ここでオートマトンの出力は制御対象に対して制御量となっており、これは一般に外変数であるから、ここではベクトル u で表わす。制御対象からは各個の出力 y が相次いで出される。

(vi) 各個の u および y から、最適化制御のための各個の目的関数 I_l [$l: 1, 2, \dots, k$] と、今までに得られたすべての目的関数の平均値 $\bar{I}$ とが計算される。

(vii) 各個の目的関数 I_l を順次平均値 $\bar{I}$ と比較して、次式により、新しい各個の MF $f_{ij,l}$ が決定・修正される。

$$f_{ij,l} = \alpha f_{ij,l} + (1-\alpha) \lambda \quad \text{--- (II)}$$

ここで、 $l: 1, 2, \dots, k$

α : MF の修正の速さに関する定数 ($0 < \alpha < 1$)

$$\lambda = \begin{cases} 1: I_l > \bar{I} \text{ のとき [これを成功という]} \\ 0: I_l \leq \bar{I} \text{ のとき [これを失敗という]} \end{cases}$$

(viii) あいまいオートマトンの MF の一部が式 (II) の結果に基づいて修正され、以後 (ii) ~ (vii) の操作が繰り返される。この場合、修正された枝による遷移はランダムではなくなる。

以上枝出力形の学習制御系の動作を説明したが、これによって次のような学習制御が行なわれていることになる。

(i) 式 (II) によって、ある枝 ij の MF f_{ij} を修正していくとき、その修正回数を n とすると、 $\lim_{n \rightarrow \infty} f_{ij} = \lambda$ となることが容易にわかる。ゆえに、式 (II) の修正動作を続けていくにつれて、成功に寄与した枝の MF は 1 に、また失敗をもたらした枝の MF は 0 にそれぞれ近づく。このことは、オートマトンのすべての枝のなかから、最適制御量に対応するもの (MF が 1 に近いもの) を選り出すためにそれ以外の枝 (MF が 0 に近いもの) を淘汰することになり、一種の自己組織動作 (Self-organization) が行なわれていることに

なる。

(ii) ありまいオートマトンの遷移を高次遷移マトリクスに基づいて行なわせることにより、局所的な最適点 (Local Optimum) や鞍点 (Saddle Point) に落ち込むことを防ぐことができる。重複の枝を取り除いたループ・パスを経て遷移させているが、重複の枝を取り除くことにより、同じ枝に対する余分な学習や、最適点でないところで一次の自己ループを繰り返して循環することを避けさせ、またループ・パスをとらせることにより、成功に寄与した枝を繰り返して行うようにして学習効果を高めるようにしている。

(b) 状態出力形

(i) ありまいオートマトンにおいては、枝のほかに状態にも MF を考え、初めに各状態の自己ループの MF を 0 に、またすべての枝および状態の MF を 0 と 1 との中央値 0.5 に定めておき、入力 x_1 によって、ある状態から任意の状態へランダムに遷移が行なわれるようにしてある。

(ii) ある状態 S_a をランダムに探索基点 (Base Point) に選ぶ、ここから他の任意の状態 S_b にランダムに遷移を行なわせ、状態 S_a および S_b に対応する制御量と制御対象の出力とから目的関数 I_a および I_b が目的関数計算部において計算される。

(iii) $I_a < I_b$ すなわち成功であれば, S_b 以外の他の状態 S_c をランダムに選んで遷移が行なわれる。以下成功である限り以上と同様の手順が続けられる。また失敗のとき (たとえば $I_a \geq I_b$, $I_a \geq I_c$ など) は, ここで S_a を探索基点とする一連の探索を終り, 状態 S_a の MF f_a および f_M までの一連の探索によって得られた目的関数のうち最大のものに対応する状態 S_M の MF f_M をそれぞれ次式によって修正して新しい MF f'_a および f'_M が定められる。

$I_a \geq I_M$ の場合

$$\left. \begin{array}{l} f'_a = \alpha f_a + (1-\alpha) \\ I_a < I_M \text{ の場合} \\ f'_a = \alpha f_a \\ f'_M = \alpha f_M + (1-\alpha) \end{array} \right\} \text{----- (12)}$$

ここで, α : MF の修正の速さに関する定数 ($0 < \alpha < 1$)

(iv) 探索されたすべての状態の間の枝の MF (一般に f_{ij} , f_{ji}) は次式によって計算された値に修正される。

$$\left. \begin{array}{l} f_{ij} = 0.5 \left[1 + \frac{I_j - I_i}{I_p} \right] \\ f_{ji} = 1 - f_{ij} \end{array} \right\} \text{----- (13)}$$

ここで, I_p : I_i, I_j をその最大値が 1 に近い正の少数になるように無次元化するための定数

(V) 現在までに遷移した状態のうちから MF の最大の状態を選び、これを次の探索基点として、以上の手順が繰り返される。このとき、枝の MF のうち式 (13) によって値の定められたものもあるので、遷移に際しては MF の最大の枝を選ぶようにする。また、式 (13) における I_i および I_j については、測定誤差や外乱の影響をできるだけ少なくするために、同じ状態（たとえば S_i ）についての m 個の目的関数値の平均値 $\bar{I}_i(m)$ を次式によって求めて用いるようにする。

$$\bar{I}_i(m) = \beta \bar{I}_i(m-1) + (1-\beta) I_i(m) \quad \dots (14)$$

ここで、 β : 平均の荷重定数 ($0 < \beta < 1$)

(Vi) 以上の手順を繰り返していったら、ある状態の MF がほぼ 1（たとえば 0.99）になると、その状態を最適点に対応するものとして、その自己ループの MF を式 (13) によって定められる値 (0.5) にし、以降繰り返し最適制御量が制御対象に加えられるようにする。

以上状態出力形の学習制御系の動作を説明したが、これによって次のような学習制御が行なわれていることになる。

各枝の MF は式 (13) によって計算・修正されて、だんだんと目的関数の応答面の勾配に対応するようになる。一方、ある状態の MF は式 (12) によって繰り返し修正されて、だんだんと最適点に対応する状態の MF は 1 に、またそれ以外の状

態の MF は 0 にそれぞれ近づく。すなわち、あいまいオートマトンに最適点の位置を強調した、目的関数のモデルが形成されることになる。オートマトンは、このモデルに基づいて、常に最大の MF の状態を探索基点として最大勾配の MF の枝を経て遷移するから、いわゆる Evolutionary Operation⁽⁶⁾ に近似でかつ一そう広域の探索を行ない、真の最適点をすみやかに探し、かつ自己ループを繰り返してその最適点を保持することができる。

4. シミュレーション実験

前述の学習制御の動作を実験的に検討するためにディジタル計算機によるシミュレーション実験を行なった。

目的関数としては次式のもの⁽⁷⁾が用いられた。

$$I(u_1, u_2) = (1 + 8u_1 - 7u_1^2 + 7/3 u_1^3 - 1/4 u_1^4)(u_2^2 e^{-u_2}) \cdots (15)$$

これは制御量が u_1, u_2 の 2 変量で、最適点のほかの一つの局部的最適点とあん点とをもっている。 $I(u_1, u_2)$ の検出に当っては、式 (15) の計算値に正規分布の乱数 (標準偏差の) に比例する測定誤差が含まれるものとする。

以下枝出力形と状態出力形とにわけて記述する。

(a) 枝出力形

この場合のシミュレーションのフロー・チャートを図 2 に、

また実験結果を 図3 にそれぞれ示す。

これらの結果から次のことがわかる。

(i) 学習の初期には、図3 (a) に点・によって例示されるような広範囲にまたがる数点によって探索が行なわれている。

(ii) 30 回ほどの広域探索ののち、最適点近くの数点について狭域探索をするが、 I が $\bar{I}$ よりも小さくなると失敗となり、失敗をもたした枝の MF が小さくなって、探索点は再び広がり最適点の広域探索を初める。

(iii) 10 回ほどの広域探索ののち、最適点を見出し、この点と準最適点との間を繰り返し遷移して定常状態に落ちつく。

(b) 状態出力形

この場合のシミュレーションのフロー・チャートを 図4 に、また実験結果を 図5 にそれぞれ示す。

これらの結果から次のことがわかる。

(i) 探索基点の移動状況を 図5 (a) でみると、局所的最適点およびあん点の付近を通過しているが、この付近を避けて真の最適点に到達している。また 図5 (b) によると、探索基点での目的関数の値が着実に大きくなるように、探索基点が推移して Evolutionary Operation に近似の動作が行な

られる。

(ii) 図5(b)によれば、一たん最適点を捕まえても、さらにそこを探索基点として他の点を広域探索して確定を期し、最後に最適点に相当する状態の自己ループを循環して、小さいハニキング・ロスで定常状態に落ち着いている。

5. おまけ

以上、あいまいオートマトンとして枝出力形および状態出力形のものを定義し、かつこれらを用いる学習制御について述べた。学習制御において、あいまいオートマトンは初めにあいまいな構造をもっているが、学習の進行につれてその構造が確定されていき、ついに最適制御量を出すようになるが、この場合、枝出力形は自己組織的な動作を、また状態出力形は Evolutionary Operation に近似の動作を行なっていることが明らかにされた。また、枝出力形では学習速度は速いが、定常状態でのハニキング・ロスが大きく、さらに各枝と制御量との組み合わせによって制御結果がかなり変化するが、一方状態出力形では定常状態でのハニキング・ロスは小さいが、学習速度が遅く、それぞれ一長一短があり、今後の問題として残されている。

6. 文献

- (1) L.A. Zadeh : "Fuzzy Sets", Information and Control, Vol. 8, No. 3 (June, 1965), pp. 338 ~ 353.
- (2) L.A. Zadeh : "Communication: Fuzzy Algorithms", Information and Control, Vol. 12, No. 2 (Feb., 1968), pp. 94 ~ 102.
- (3) 浅居・北嶋・谷口・平井 : "あいまいオートマトンとその学習制御への応用", 電気関係学会関西支部連合大会論文集, 7-4 (昭43-11).
- (4) H. Hirai, K. Asai and S. Kitajima : "Fuzzy Automaton and Its Application to Learning Control Systems", Memoirs of the Faculty of Engineering, Osaka City University, Vol. 10, (Dec., 1968), pp. 67 ~ 73.
- (5) 浅居・北嶋・谷口・平井 : "あいまいオートマトンによる学習制御", 電気四学会連合大会論文集 (昭44-3 発表予定).
- (6) G.E.P. Box : "Evolutionary Operation - A Method for Increasing Industrial Productivity", Applied Statistics, Vol. 6, No. 2 (1957), pp. 3 ~ 23.
- (7) H.E. Zellnik, N.E. Soudart and R.S. Davis : "Gradient Search Optimization", Chemical Eng. Prog., Vol. 58, No. 8 (August, 1962), pp. 35 ~ 41.

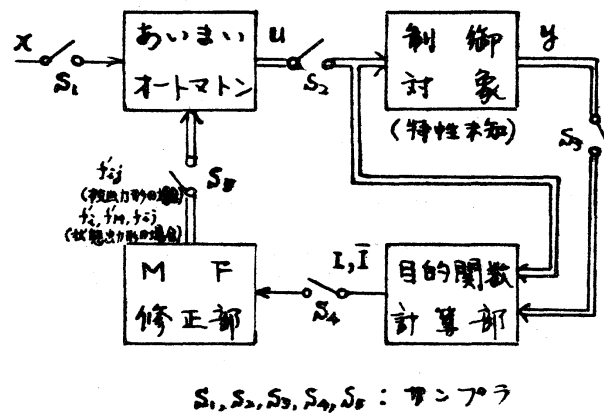


図1. 学習制御系の構成図

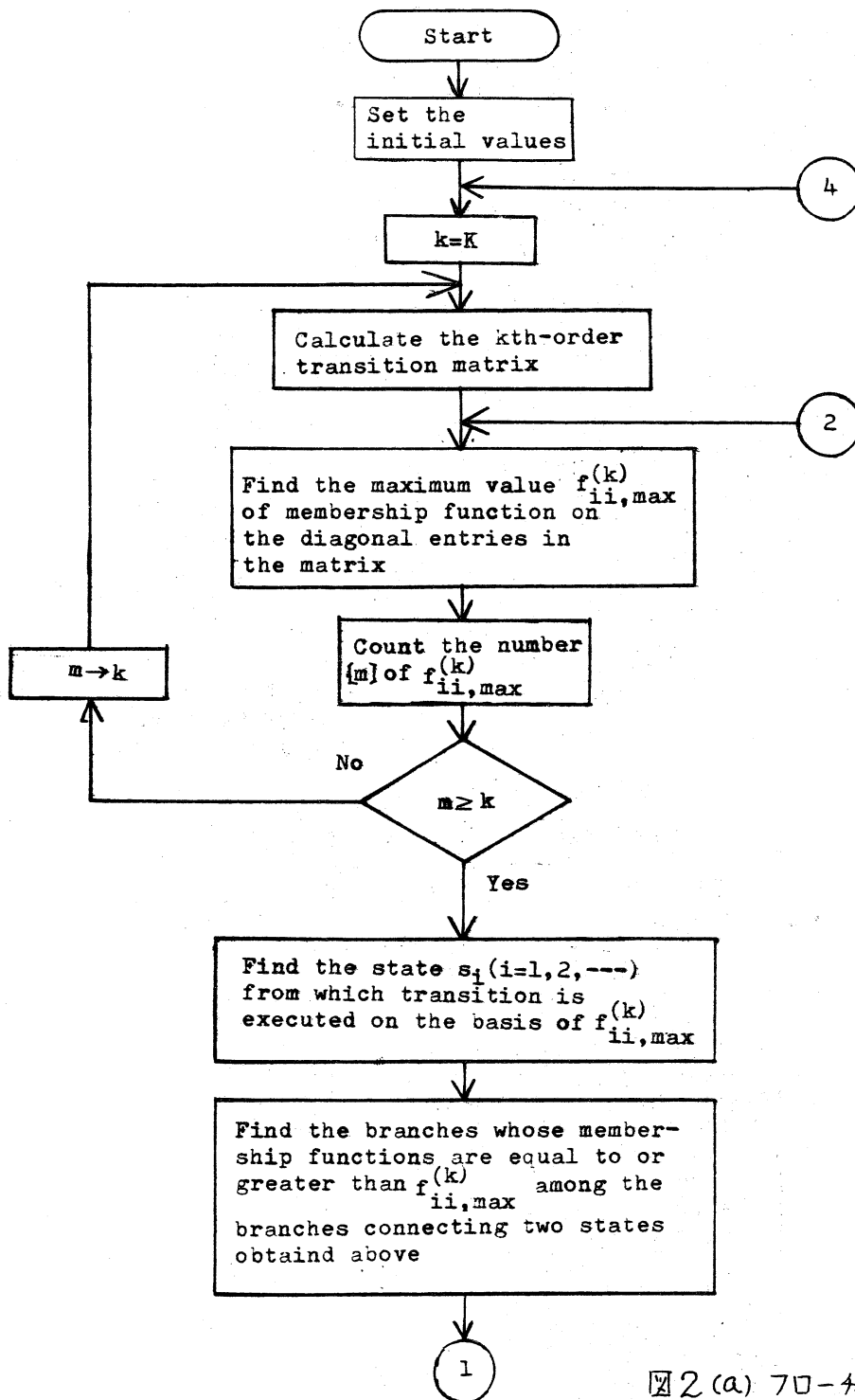
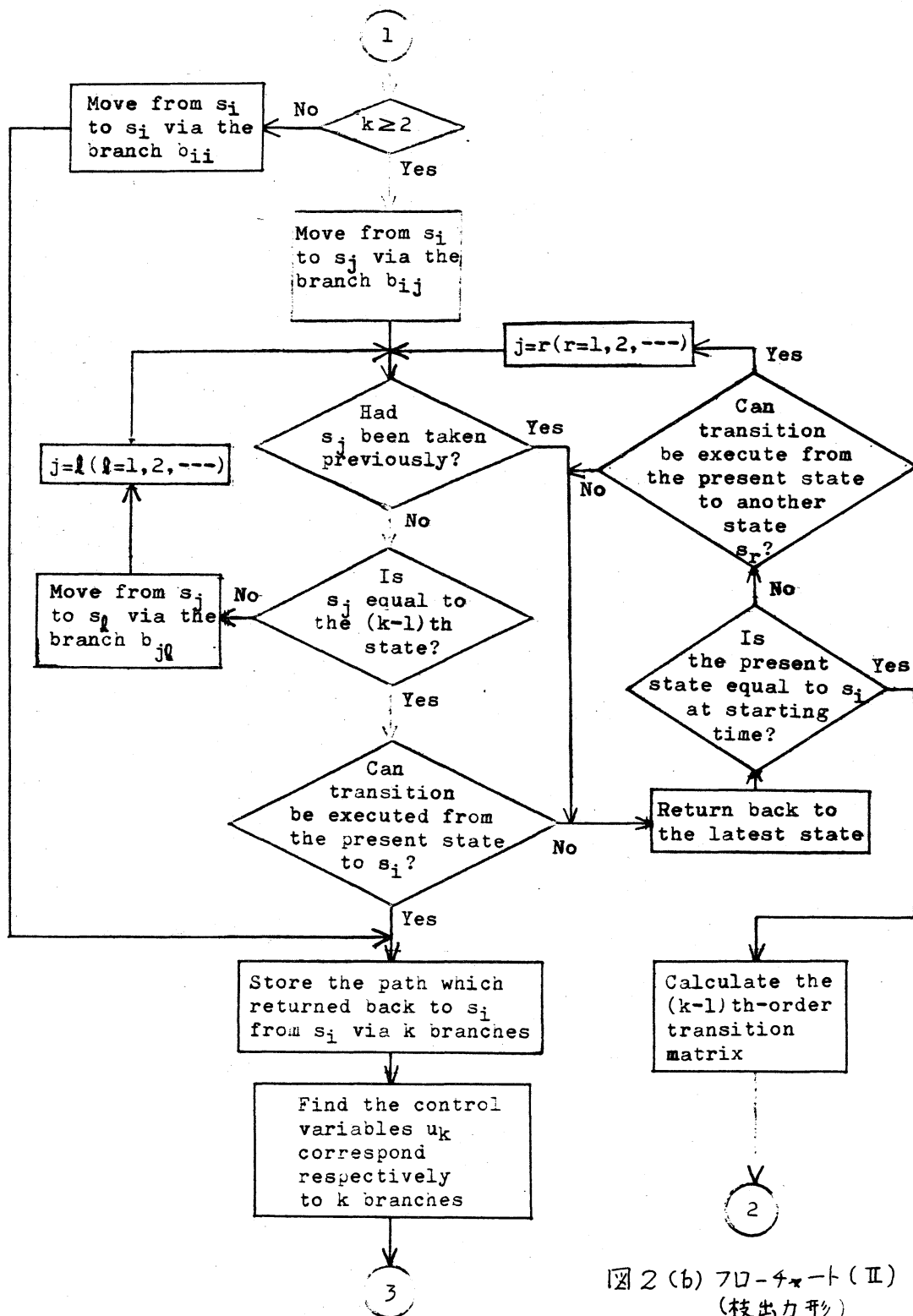


図2(a) 70-4-1(I)
(枝出力形)



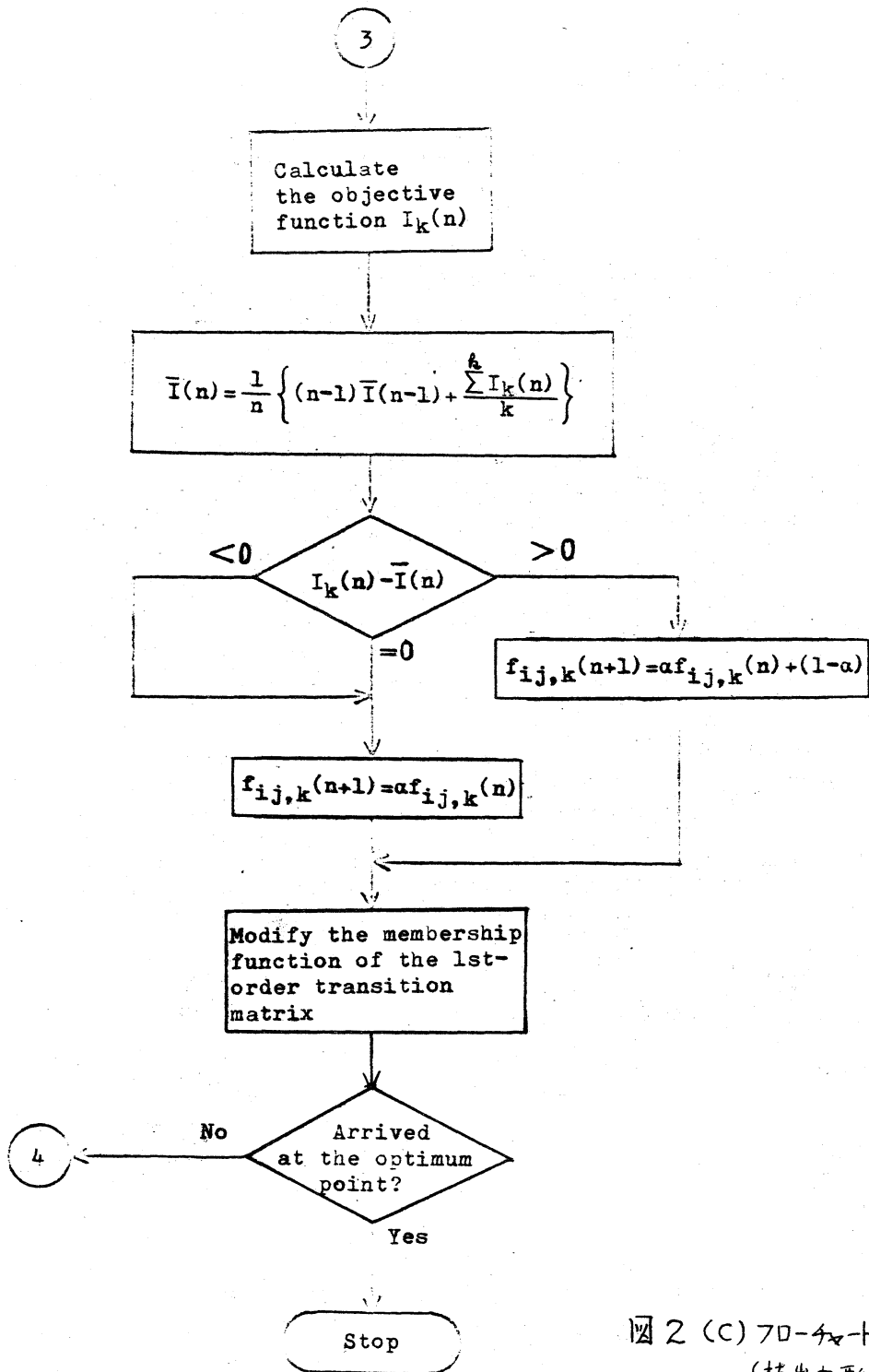
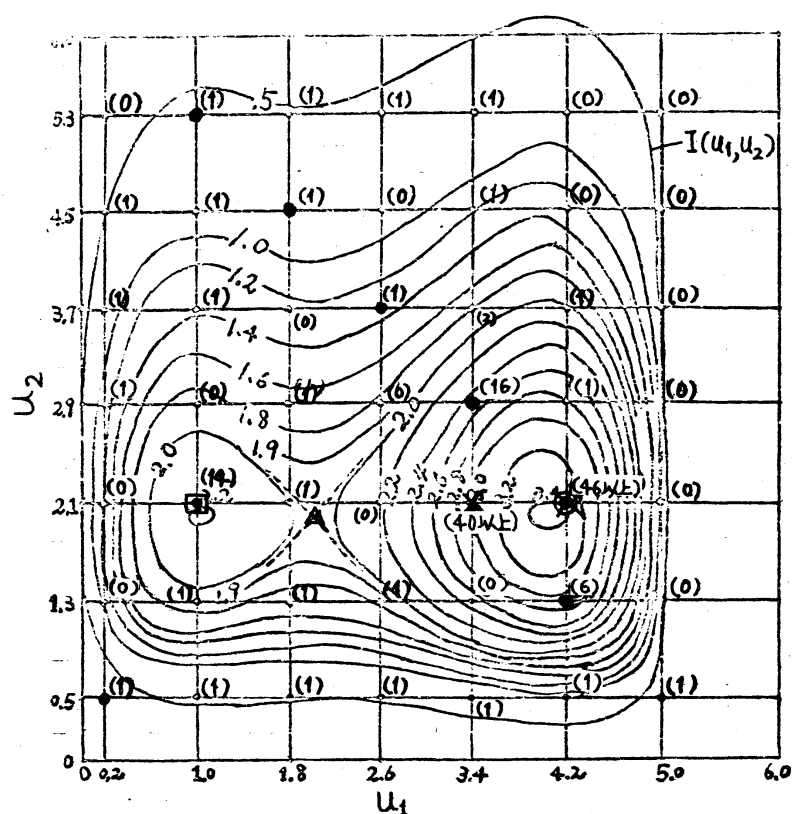


図 2 (C) フロート(Ⅲ)
(枝出力形)



●: 出発時の探索点 (7点) ×: 最終の探索点 (2点)
 ⊙: 最適点 □: 局所的最適点 △: あん点
 (): 探索の繰り返し回数 (a)

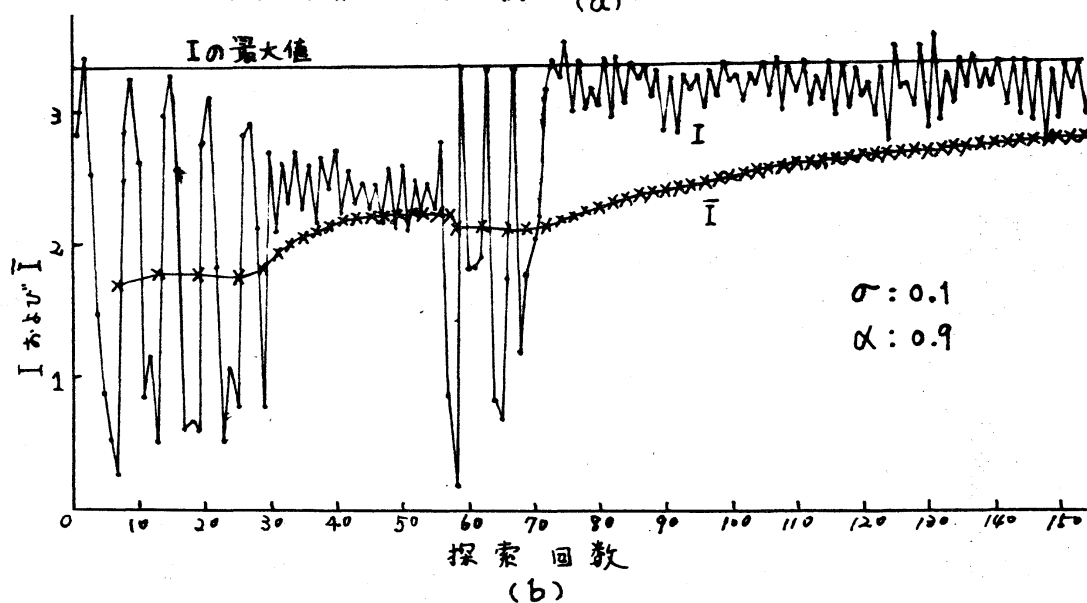


図3 シミュレーション実験の結果 (枝出カ形の場合)

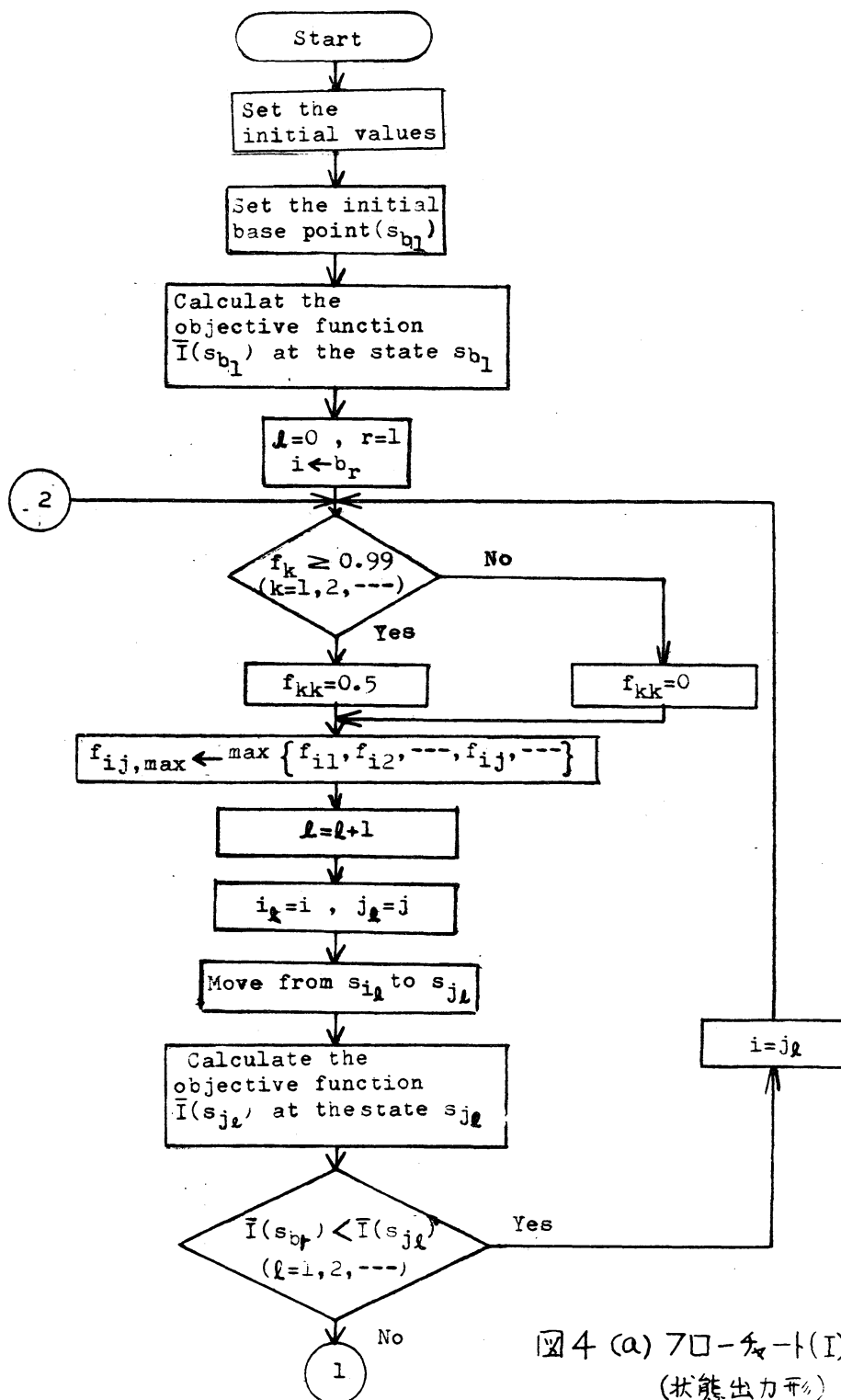


図4 (a) 7口-系-(I)
(状態出力系)

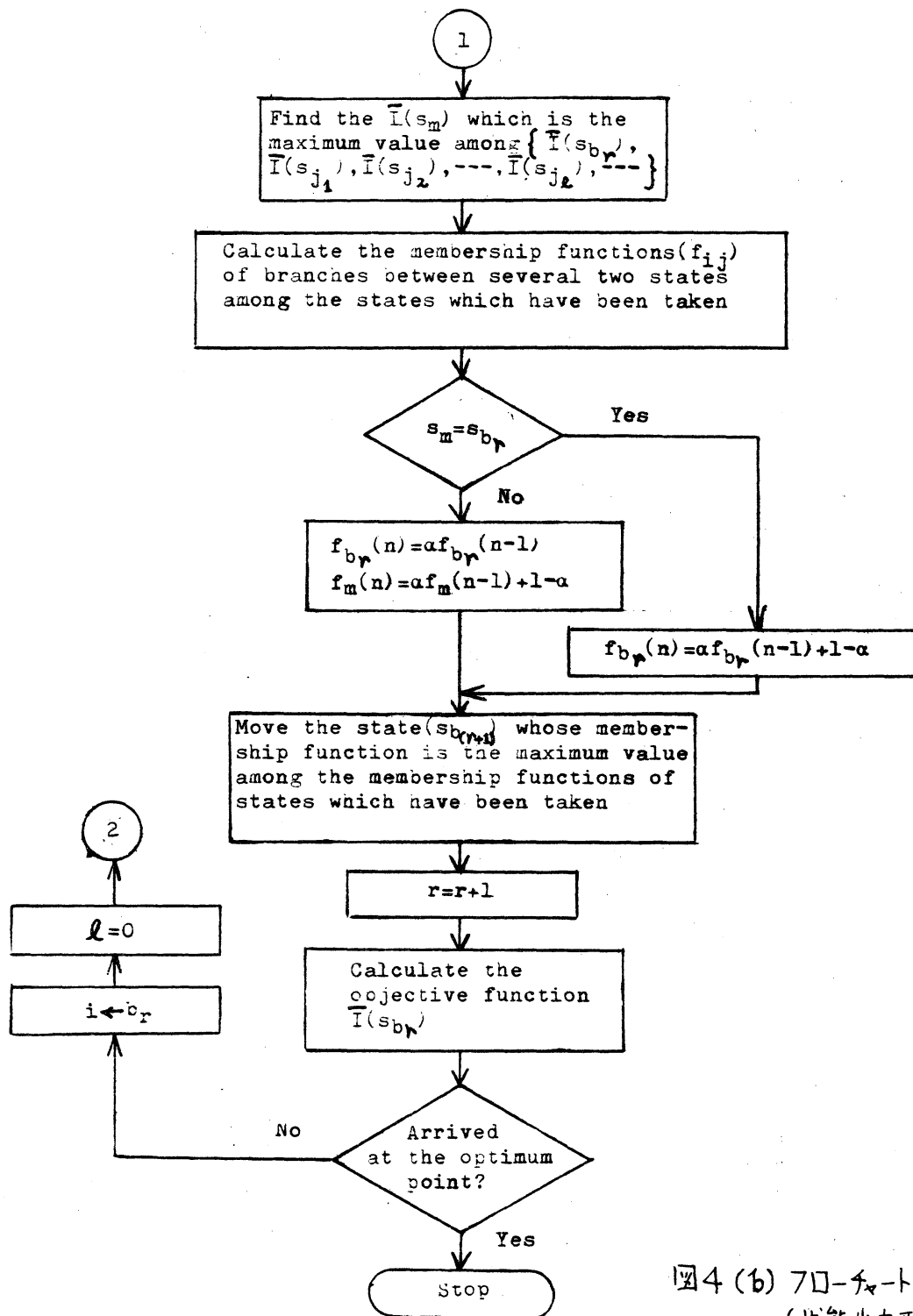
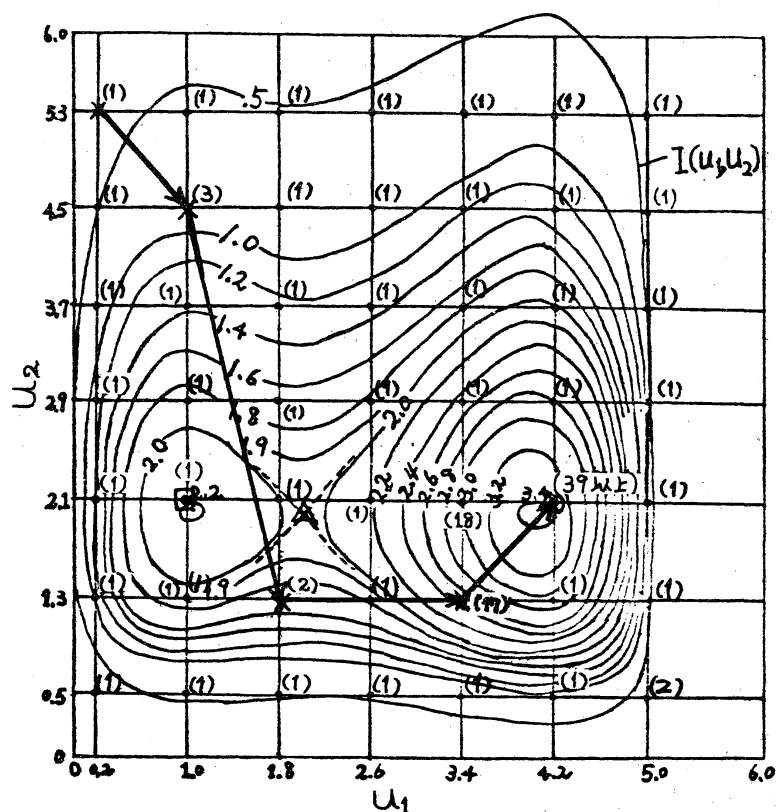


図4 (b) フロート(Ⅱ)
(状態出力形)



x: 探索基点, ⊙: 最適点, □: 局所的最適点, △: あん点
矢印: 探索基点の移動を示す, (): 探索の繰り返し回数

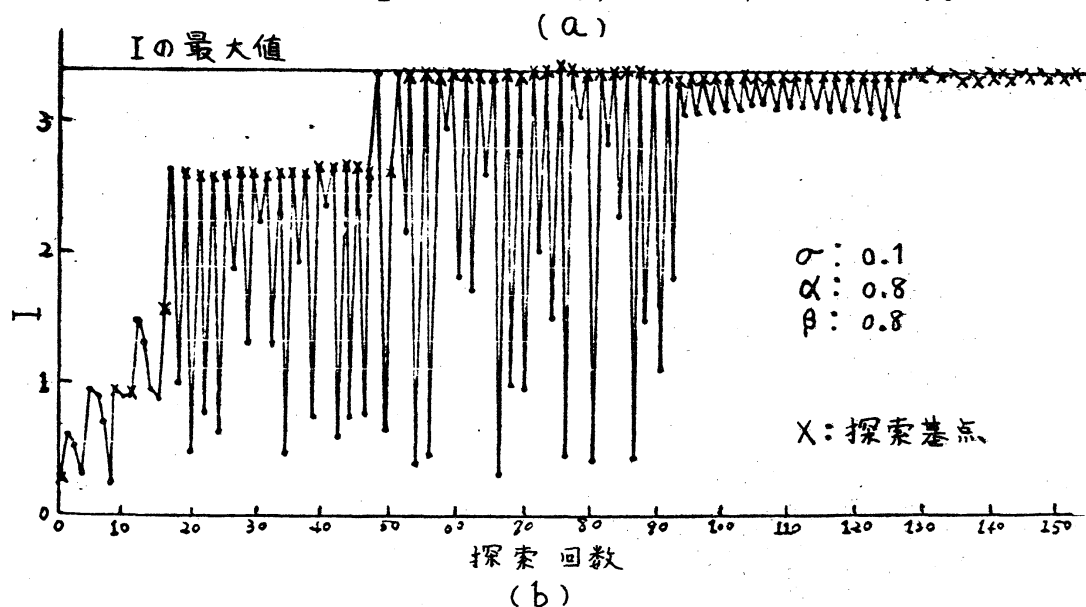


図5 シミュレーション実験の結果 (状態出力形の場合)