論文

GAを用いた
無限多値近似推論エンジンの構成[†]

山本 喜則*

以前、無限多値しきい値関数を新たに定義して近似推論エンジンを構成する方法を提案した。方法の本質は規則表が表わす超曲面を無限多値しきい値関数の式で一括表現することであり、制御への応用も有効であることを示した。本論文は、その構成において未解決の問題：非線形や非unateな規則表等を一括表現する方法を議論する。前論文[10]で述べたように、対象の超曲面はその(整数入出力に対応する)格子点が表わす有限多値論理関数をまず求め、次いでアナログ値入力を与えることで近似実現できる。したがって問題は任意の多値論理関数のしきい値関数による実現問題に帰着するが、このことは多値論理において必ずしも簡単な方法が得られているわけではない。本論文では遺伝的アルゴリズムとしきい論理で今まで得られている知見を活用して多段回路網を自動生成する方法を述べる。方法の要点は、多値に対応した重みとしきい値より成る遺伝子型を考え、最大重みを生成するよう初期値設定した遺伝子型に単純GAのアルゴリズムを適用する点にある。前論文で例示した複雑な規則表もこの方法で回路網実現できる。併せて本手法の能力と限界について考察する。

キーワード：多値論理、しきい値関数、多値論理関数のしきい値素子実現、しきい論理、ファジィ推論、遺伝的アルゴリズム、ニューラルネット

1. はじめに

多値しきい値関数の連続論理への一つの拡張として無限多値しきい値関数を以前[1,2]で定義した。それによると推論規則が表す超曲面(制御面)を無限多値しきい値関数の式で一括近似表現することでファジィ推論と同様な処理がより高速かつ簡単に実現できる。しかし、複雑な超曲面は単一の無限多値しきい値関数で表せないため、複数の無限多値しきい値関数合成の方法が必要となった。

[3]では、多重のしきい値を持つ拡張無限多値しきい値関数を新たに定義し、それを用いた一つの一般的解法を示したが、これは無限多値しきい値関数の合成ではない。一方、[10]において、超曲面が表す連続関数を無限多値しきい値関数の多段結合で近似する理論的基礎を述べた。そこでの結論は、まず連続関数の整数値入出力に対応する格子点が表現する多値論理関数を考え「多値論理関数の通常の多値しきい値関数による合成問題」を解決して得られた回路網にアナログ値入力を与えれば目的の連続関数の近似表現を得ることができるというものであった。

しかしながら、多値しきい値関数は関数的に完全であるから理論上合成可能であるものの、この合成実現の問題は完全に解決されているわけではない。2値論理

の場合様々な方法が提案されたが[4,5]、多値では情報量がきわめて多く複雑で、しきい値素子の万能性が一時期ほど注目されないなかでは、この方面の研究は少ない(たとえば、[6][7]など)。これらの研究にみられる方法は、論理的であるが理解が難しくまたコンピュータ向きではない。その後、こうした研究は見あたらない現状である。

一方、しきい値素子は神経回路網の構成要素として前述の合成とは意図が異なる用い方をされ、BP法という有力な方法が定着している。しかも、神経回路網においては、4層ネットワークで中間層のユニットを必要なだけたくさん使えば、ユニット間の結合とシグモイド関数のしきい値を適切に設定することで任意の連続関数： $[0,1]^n \rightarrow (0,1)$ を近似できるとされている[9]。この立場で考えれば、すでに[3][10]で触れたように、一般に超曲面をしきい値素子の多段結合で実現することは原理的に可能だといえる。

しかしながら、その方法は教師信号との乖離の総和が最小である回路網を見いだすものであって、格子点での一致を求める本論文の多段実現には必ずしも適さない。またシグモイド関数の代わりに無限多値しきい値関数を用いて同様な結論が言えるかは不明である。

本論文では任意の多値論理関数の多値しきい値関数による合成実現の手法を検討する。この主題を解決するために、構造(ユニット個数と結合の仕方)の決定、重みとしきい値の修正に遺伝的アルゴリズムを用いる。多値論理関数の多値しきい値関数による合成実現は格子点における出力が教師信号と完全に一致するフィードバックループのない神経回路網を構成することともいえるが、神経回路網の構成に遺伝的アルゴリズム

[†] A Synthesis of an Infinitely Multi-valued Approximate Reasoning Engine by Means of Genetic Algorithm
Yoshinori YAMAMOTO

* 高崎経済大学経営学科
Dept. of Business Administration, Takasaki City University of Economics

を用いる試みは、Montana and Davis[16]をはじめ、長尾[15]、北野[17]など多くの研究がある。しかし、完全一致の回路網でない点、多値入出力では正解に至らない恐れがある点など、上記の主題にそのまま適用できるものではない。

本論文は、第28回多値論理国際シンポジウムにおいてこの主題を扱った予稿原稿[19]に、しきい論理の研究で得られている知見を利用する手法およびその他の実験例を加筆し全体を修正したものである。前論文で提起された非線形・非unateな制御規則を新しい方法で実現実験した結果を示す。得られた回路網にアナログ値入力を与えることで、冒頭に述べた近似推論エンジンを実現することができる。併せてこの手法の能力と限界を考察するため、ランダムに生成した多値論理関数を実現するいくつかの実験例を示す。

2. 定義, 性質

次の演算を定める. ここで, $\mathbf{x}_1 \in [a_1, b_1]$, $x_2 \in [a_2, b_2]$
(a_1, b_1, a_2, b_2 は実定数, $a_1 < b_1, a_2 < b_2$)

論理和 $\vee : x_1 \vee x_2 = \max\{x_1, x_2\}$

論理積 $\wedge$: $x_1 \wedge x_2 = \min\{x_1, x_2\}$

否定⁻: $x_i = a_i + b_i - x_i$

(+ : 算術和, - : 算術差, $i=1,2$)

算術和 $+$: $x_1 + x_2$

$$\text{算術積} \cdot : x_1 \cdot x_2 \quad (1)$$

演算の優先順序は、算術積、否定、論理積、論理和、算術和の順とする。ただし、演算の順序はかっこが優先する。この演算系に基づく公式については文献[8]に詳しい。算術積の記号はこの後省略する。

[定義1][1,2] 全ての実数の集合を R とする. R の部分閉区間 $R_i (i=0, 1, \dots, n)$ を考える. 変数 $x_i (x_i \in R_i)$, 定数 ($\in R$), 及び式(1)の演算の有限回の組合せによる式が表現する $R_1 \times \dots \times R_n$ から R_0 への連続写像を, n 変数無限多値関数という.

[定義2][1,2] f は定義1の無限多値関数であるとする. $A, A' \in R_1 \times \cdots \times R_n$ は第 i 成分のみが異なるとする. f がこのような任意の $A = (a_1, \cdots, a_n)$, $A' = (a_1', \cdots, a_n')$ に対して

$$a_i \leq a_i' \text{ ならば } \begin{matrix} f(A) \leq f(A') \\ (\geq) \end{matrix} \quad (2)$$

を満たすとき, f は変数 x_i に関して単調増加(単調減少)であるという. 全ての変数について単調増加か減単調少であるとき, f は unate であるという. さらに全ての変数について単調増加であるとき, f は単調であるという.

[定義3][1,2] 定義1の n 変数無限多値関数 $f: R_1 \times$

$\cdots \times R_n \rightarrow R_0$ ($R_i \subseteq R$, $i=0, 1, \cdots, n$) を考える. f の関数値が, ある実定数 $w_1, \cdots, w_n$, $t_1, \cdots, t_m$ (m : 自然数, $t_1 \leq \cdots \leq t_m$) に対して式(4)のように定まる時, f は重みベクトル $W = (w_1, \cdots, w_n)$, しきい値と関数値の集合 $\{T, f(T)\} = \{(t_1; f(t_1)), \cdots, (t_m; f(t_m))\}$ を持つ無限多値しきい値関数であるという. ここに, WX は $W = (w_1, \cdots, w_n)$ と $X = (x_1, \cdots, x_n)$ の内積である. また, $f_0, \cdots, f_m$ は,

$$f_i(t_{i+1}) = f_{i+1}(t_{i+1}) \quad (i=0, 1, \cdots, m-1) \quad (3)$$

を満たし WX を変数とする, 単調増加な, R から R への1次以下の, 互いに異なる1変数関数である.

$$f(x_1, \dots, x_n) = \begin{cases} f_0(WX) = \inf R_0 & (WX \leq t_1 \text{ のとき}) \\ f_1(WX) & (t_1 \leq WX \leq t_2 \text{ のとき}) \\ \cdots\cdots\cdots & \\ f_i(WX) & (t_i \leq WX \leq t_{i+1} \text{ のとき}) \\ \cdots\cdots\cdots & \\ f_m(WX) = \sup R_0 & (t_m \leq WX \text{ のとき}) \end{cases} \quad (4)$$

[定理1][1,2] f は定義3の無限多値しきい値関数であるとする. f は次式のように表現できる.

$$f(x_1, \dots, x_n) = f_0 \triangle f_1 \triangle f_2 \triangle \dots \triangle f_m \quad (5)$$

ここで、記号 $\triangle$ は、両端の f_i と f_{i+1} が上に凸のときは演算子 $\wedge$ 、下に凸のときは演算子 $\vee$ を表す。

(例1) $R = \text{区間}[0, 2]$ のとき, $R \times R \rightarrow R$ なる無限多値しきい値関数 f を考える. 重みベクトルを $W = (w_1, w_2) = (-1, 2)$, しきい値と関数値の集合を $\{T; f(T)\} = \{(-1; 0), (0; 1), (3; 1), (4; 2)\}$ とすると, f の出力は式(6)のように定まる.

$$f(x_1, x_2) = \begin{cases} f_0(WX) = \inf R_0 = 0 & (WX \leq -1 \text{ のとき}) \\ f_1(WX) = WX + 1 & (-1 \leq WX \leq 0 \text{ のとき}) \\ f_2(WX) = 1 & (0 \leq WX \leq 3 \text{ のとき}) \\ f_3(WX) = WX - 2 & (3 \leq WX \leq 4 \text{ のとき}) \\ f_4(WX) = \sup R_0 = 2 & (4 \leq WX \text{ のとき}) \end{cases}$$

ここに, $WX = -x_1 + 2x_2$. (6)

ここに, $WX = -x_1 + 2x_2$.

WX と f の関係を図1に示す. f は区間 $[0, 2]$ の任意の

図1 例1の無限多値しきい値関数

実数値をとる関数である。\$f\$ の式は式 (5) (6) により以下のよう表せる。

$$\begin{aligned} f(x_1, x_2) &= f_0 \vee f_1 \wedge f_2 \vee f_3 \wedge f_4 \\ &= -1 \vee (WX + 1) \wedge 1 \vee (WX - 2) \wedge 2 \\ &= -1 \vee 1 \wedge (-x_1 + 2x_2 + 1) \\ &\quad \vee 2 \wedge (-x_1 + 2x_2 - 2) \end{aligned}$$

[系1][1,2] 定義3の無限多値しきい値関数は、変数 \$x_1, \dots, x_n\$ に関して unate である。しかも、次の関係が成り立つ。

$$\begin{aligned} w_i > (<) 0 &\Leftrightarrow f \text{ は } x_i \text{ に関して単調増加(減少)} \\ w_i = 0 &\Leftrightarrow x_i \text{ は冗長。} \end{aligned}$$

[定理2][1,2] 無限多値しきい値関数は離散多値しきい値関数を、その特殊な場合として含む。

3. 前論文から継続する問題

3.1 無限多値しきい値関数を用いた近似推論法

複数のファジィ推論規則の集まりから一つの無限多値関数を得る手続きは文献[1]に詳しく記したので、ここでは簡単にふれる。

各々のファジィ推論規則は式 (7) の形式で表されているとする。

$$\begin{aligned} A_1 \text{ かつ } A_2 \text{ かつ } \dots \text{ ならば } B \\ (A_1, A_2, \dots, B \text{ はファジィラベル}) \end{aligned} \quad (7)$$

2入力の場合、たとえば表1のような推論規則表で表せる。NB, ZO, PB はそれぞれ Negative Big, Nearly Zero, Positive big を表すが具体的なメンバーシップ関数は不要である。

近似推論は以下の手順で行う。

Step1) ファジィラベルを整数値に置き換える。数値は元のラベルが示す大まかな大小と論理的に対応していれば何でもよい。これにより crisp 値の入出力表が得られる。もし幾つかの出力値が未定義のときは、周辺の出力値から推測して決定する。

(例2) 表1の NB,ZO,PB をそれぞれ0,1,2に置き換えると表2を得る。

Step2) この関数表を一つの(定義1の)無限多値関数の不完全定義関数表と考える。この関数表に記された全ての入出力の組に対して同一の値を出力する無限多値しきい値関数が存在するならば、その重み、しきい値、式表現を求める。

(例3) 表2を不完全定義無限多値関数と考え、表中すべての入出力が一致するような定義3の無限多値しきい値関数を考える。重みベクトル \$W = (-1, 2)\$, しきい値と関数値の集合 \$\{T, f(T)\} = \{(-1; 0), (0; 1), (3; 1), (4; 2)\}\$ のとき一致し (\$WX\$ の変化を表3に示

表1 ファジィ推論規則表

x_2 x_1			
	NB	ZO	PB
NB	ZO	ZO	PB
ZO	NB	ZO	ZO
PB	NB	ZO	ZO

表2 数値ラベル付推論規則

x_2 x_1	0	1	2
	1	1	2
0	1	0	1
1	0	1	1
2	0	1	1

表3 \$WX\$ の変化

x_2 x_1	0	1	2
	0	2	4
0	0	2	4
1	-1	1	3
2	-2	0	2

単一の無限多値しきい値関数またはその多段結合

図2 推論の基本ブロック図

す), \$f\$ は,

$$\begin{aligned} f &= -1 \vee (WX + 1) \wedge 1 \vee (WX - 2) \wedge 2 \\ &= -1 \vee 1 \wedge (-x_1 + 2x_2 + 1) \vee 2 \wedge (-x_1 + 2x_2 - 2) \end{aligned}$$

で表される。これは例1で取り上げた関数である。

Step3) 得られた無限多値しきい値関数 \$f\$ の式から図2のようにして推論結果を得る。式で書けば(8)のようになる。ここで \$^0x_1, \dots, ^0x_n\$ は推論のための現実の crisp な入力情報、\$^0z\$ はそのときの現実の crisp な推論結果である。\$\lambda_i (i=1, 2, \dots, n)\$, \$\phi\$ は入出力値調整のためのゲイン関数である。

$$\begin{aligned} ^0z &= \phi(z) = \phi(f(x_1, \dots, x_n)) \\ &= \phi(f(\lambda_1(^0x_1), \dots, f(\lambda_n(^0x_n))) \end{aligned} \quad (8)$$

3.2 問題点

無限多値しきい値関数による推論法は、ファジィ推論規則表を一つの無限多値しきい値関数の式で表せるかどうかが要点である。ところが現実のファジィ規則表には以下のようなものが存在する。まず、次の確認をしておこう。

離散入力に対する出力値が記された関数表を考える。この関数表において、\$x_i\$ が増加(減少)ならば \$f\$ の値が増加(減少)するとき、表は単調増加(減少)であるとい

う、表がすべての変数について単調増加か減少であるとき、表はunateであるという。

この関数表のような入出力の組合せを与える無限多値関数は数多く存在する。今、表よりなんらかの方法で見いだされた関数 f が定義1の無限多値関数であるとする。表がunateならば、これらの f のうち少なくとも一つはunateであるものが存在する。なぜならば、見いだされたすべての f が非unateならば表はunateになり得ないからである。

[様々な推論規則表]

- 1) unate だが非線形な変化を含む規則表。
- 2) unate でない規則表。
- 3) 多入力の場合の規則表。

1)は、表4(a)のようにunateだが、非線形な変化を含む場合である。重みと入力の線形式 WX (荷重和)だけでは曲線的な変化を表現できない。

2)は、表4(b)のような場合である。変化の様子はunateでない。無限多値しきい値関数はunateなので、明かに単一なものでは表現できない。

3)は、規則の一部しか公表されていない事例が多く良い例が見あたらないが、3次元以上の表になるため文章で表現される場合である(注：ここで「多」は3以上を意味する。多値論理の場合と同様)

以上、1)–3)は単一の無限多値しきい値関数では表現できない。

3.3 解決のための方策

以上の問題点について、文献[3]で検討した。それらは次のようにまとめられる。

方策1: 非線形な変化をする入力ゲイン関数を、本来の入力ゲイン関数とファジィしきい値関数の間に入れるなど、入力ゲイン関数を工夫する。出力ゲイン関数についても同様。

方策2: 多重しきい値を持つ無限多値しきい値関数を新たに定義し、「普遍重み」を用いて一つの関数で表す。ここで普遍重みとは次のものを指す。
「ファジィラベルを一方に整数値 $0, 1, 2, \dots$ で置き換えたとき、変数 x_i のラベルを b_i とする。重みを、 $w_i = b_{i+1} \cdot b_{i+2} \cdot \dots \cdot b_n$ ($i=1, 2, \dots, n-1$)、 $w_n=1$ と定める。」

方策3: 無限多値しきい値関数を多段結合する。

方策1は、非線形な規則表の変化に合うようなゲイン関数を見いだすことが要点であるが、どのように同定すればよいのかの組織的な手法の発見はかなり困難である。

方策2は現在までのところ実現可能な唯一の方法で

表4 単一のファジィしきい値関数で表しにくい規則表

(a)		(b)	
x_1	x_2	x_1	x_2
	NB NM ZO PM PB		NB NM ZO PM PB
NB	NB NB NM ZO ZO	NB	ZO PM PB PM ZO
NM	NB NM NM ZO ZO	NM	NM ZO PM ZO NM
ZO	NM NM ZO PM PM	ZO	NB NM ZO NM NB
PM	ZO ZO PM PM PB	PM	NM ZO PM ZO NM
PB	ZO ZO PM PB PB	PB	ZO PM PB PM ZO

PB: Positive big PM: Positive medium ZO: Nearly zero
NM: Negative medium NB: Negative big

あり、具体的な手順が文献[3]で与えられた。しかし、二つの問題が残る。一つは、普遍重みを用いると最大で「出力値の個数-1」個のしきい値が発生するため、得られる関数の式が大きくなるおそれがあることである。もう一つは、得られた式中に WX 中とは別に算術演算子が現れることである。これは定義3から本質的に起こり得ることであるが、素子をハードウェア実現する上では好ましいことではない。

方策3は、神経回路網研究と関わりがある。神経回路網情報処理の研究において、次の事実が知られている。
[連続関数近似についての完全性][9]: それぞれのユニットがシグモイド関数によるしきい値素子である4層ネットワーク(学習を行うユニットは実質2段)において、中間層のユニットを必要なだけたくさん使えば、ユニット間の結合、およびシグモイド関数のしきい値を適切に設定することによって、任意の連続関数 $f: [0, 1]^n \rightarrow (0, 1)$ を任意の精度で近似することができる。

この事実は、先の三つの問題点を含む超曲面であっても理論的には、シグモイド型しきい値素子の多段結合(神経回路網)が表わす連続関数で近似できることを意味する。ところが、有力なBP法などでも回路実現のゴールは教師信号との乖離の総和が最小なものを見出すことであって必ずしも完全一致を目指していない(実際、シグモイド関数を用いる限り、完全一致は困難であろう)。

本論文の近似推論法に方策3を検討する際の留意点:

一方、著者が[1]–[3]で提案してきた近似推論法は、まず与えられた推論規則は確定しているものと仮定し、新たに定義した関数を用いた通常ファジィ推論の代替方式を提案するものである。もとよりファジィ推論規則自体が厳密なものでなく近似的なものであることは知られているが、提案のポイントはファジィ推論規則全体を無限多値しきい値関数の組み合わせでどのように近似表現するかということにある。また、推論エ

ンジンを表現する連続関数は、元のファジィ推論規則のラベルに対応する整数入出力値[1](格子点)においては、その入出力値と完全一致することを求める。それは、少なくともそれら格子点の入出力値は与えられた推論規則・制御規則の確定的な部分を表現しているという前提に立っているからである(この立場は[1]-[3][10]でも一貫して述べてきた)。そして、これら元のファジィラベルに対応する格子点では完全一致を求めるが、それら以外の入出力値では、必ずしも元の規則の場合と一致する必要はなく、これが近似推論法と称する理由でもある。ここで重要なことは、以上の前提に立てば全体として乖離の総和は最小であってもある格子点で乖離が大きい場合、それを用いた推論値はその点および周辺で本来のものと大きく食い違うものとなり、元の推論・制御規則の確定部分を表現したといえなくなる点である。つまり、通常の教師信号との乖離の総和が最小なものを見いだす手法では上の要請は満たせないともいえよう。

ただし、シグモイド関数の代わりに無限多値しきい値関数を用いても連続関数近似についての完全性の事実が成立するかは今後も議論の余地があろう。

以上の考察を踏まえ、本論文は方策3のための新たな方法を検討する。3.1で述べた手順から知られるように、考えている格子点は有限個でありまた教師信号と完全一致することを求めるから、この問題は多値論理において「多値論理関数の多値しきい値関数による合成問題」といわれたテーマでもある。冒頭述べたように、現在この方面の研究は行われていないようである。著者は文献[10]で、有限多値しきい値関数を多段結合して得られた回路網に連続値入力を与えて所求の(格子点が整数値入出力に対応する)無限多値連続関数を近似合成できること、およびその数理的側面を多値論理関数合成問題の延長として検討した。ただし、[10]でも新たな方法は得られていない。

以下では、「遺伝的アルゴリズムを用いて、有限多値しきい値関数の多段結合により任意の多値論理関数を実現する方法」を検討する。

4. 遺伝的アルゴリズムの応用

4.1 対象とする回路網

遺伝的アルゴリズム(GA)の適用を考えている回路網は n 入力1出力で図3のブロック図のようになる。図は2入力、総素子数6の場合を例示している。各素子間の経路は実際には存在しないもの(重み0)もある。各素子への入力線の番号は、この後の表5の通し暗号の一部である。

p 値 n 変数しきい値関数は、次のように定められ

る。

[定義4] [18] 以下のように出力値が定まる n 変数 p 値論理関数 $f(X)$ は、重み $w_1, \dots, w_n$ 、しきい値 $t_1, \dots, t_{p-1}$ を持つ p 値しきい値関数であるという。ここに、 $WX = w_1x_1 + \dots + w_nx_n$ であり、 $t_1 \leq t_2 \leq \dots \leq t_{p-1}$ である。

$$\begin{aligned} f(X) &= p-1 && \text{if and only if } WX > t_{p-1} \\ f(X) &= j-1 && \text{if and only if } t_j \geq WX > t_{j-1} \\ f(X) &= 0 && \text{if and only if } t_1 \geq WX \end{aligned} \quad (9)$$

3値しきい値関数は必ず整数重みを持つことが証明されている[12]。その後一般 p 値での報告はないが、[12]での証明は本質的に3値以上にも拡張可能なものである。よって以下、整数重み(したがって、しきい値も整数)を仮定する。

回路網における素子の結合状態を表5に示す。このような表は[15]で長尾らが用いているが、素子は本質的に2値しきい値素子であり、また重みは0, ± 1 に限定されている。重みの候補となる整数値を何にするかは、このあと述べるように実はしきい論理の立場では極めて重要である。

表は、図3の4値2入力総素子数6の場合である。数字はその行の素子への(入力線の)重みとしきい値のための通し番号であり、ゴシック字部分に重み、イタリック字部分にしきい値を設定する。たとえば、通し番号

図3 多値論理関数を実現するしきい値素子回路網

表5 図3の回路網における素子の結合状態

		from 素子の番号							
to		u_1	u_2	u_3	u_4	u_5	6	7	8
		u_1	u_2	u_3	u_4	u_5	6	7	8
素子の番号	$x_1 = u_1$	×	×	×	×	×	×	×	×
	$x_2 = u_2$	×	×	×	×	×	×	×	×
	u_3	1	2	3	4	5	×	×	×
	u_4	6	7	8	9	10	11	×	×
	u_5	12	13	14	15	16	17	18	×
	u_6	19	20	21	22	23	24	25	26

3, 4, 5 には4値しきい値素子 u_3 のしきい値 t_1, t_2, t_3 を設定する。ただし各素子について、しきい値が $t_1 \leq t_2 \leq \dots$ を満たすように設定しなければならない。

4.2 遺伝的アルゴリズム適用としきい値論理の知見の応用

遺伝的アルゴリズムを適用する基本的な処理の大きな流れは図4のようになる。手順の詳細は以下の通りである。

手順1) 実現対象関数の定義：

集団サイズ、(入力変数も含んだ)総素子数は実験毎に設定する。 n 変数 p 値入力のすべての組に対して論理関数 $f(X)$ の出力値を定め、実現対象の一つの p 値論理関数を定義する。

手順2) 遺伝子型の初期設定：

一つの遺伝子型は各しきい値素子の重みとしきい値よりなる。たとえば、4値2入力総素子数6の回路網では表5の通し番号1から26に重みとしきい値に対応する整数値を設定する(遺伝子長=26)。ここで表の総素子数を $unit$ で表わす。表を一般化すれば容易に知れるように、 p 値 n 入力で総素子数 $unit$ 個の場合、一つの関数のしきい値は $p-1$ 個なので遺伝子長は次式で与えられる。

$$\begin{aligned} \text{遺伝子長} &= \sum_{j=n}^{unit-1} (j+p-1) \\ &= (unit+n+2p-3)(unit-n)/2 \quad (10) \end{aligned}$$

例3に表5の回路網の一つの遺伝子型の例を示す。

図4 多値論理関数を実現する各素子の重みとしきい値探索の手順

(例3) 4値2変数、総素子数6の回路網における遺伝子型の例：

素子番号：	1	2	3	4	5	6	7	8	9	10	11	12		
	-1	6	-1	5	9	1	5	-2	0	0	3	-4		
	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	-3	2	2	1	8	13	7	-3	0	0	3	-3	-1	0

変数 x_i の重みを w_i とすると、 w_i をいくつに初期値設定するかは p の値に関わりがあり重要である。

[重みの設定におけるしきい値論理の知見とその応用]

たとえば3値しきい値論理では3変数までの標準関数の最適構造(重みとしきい値)がすべて求められているが[13,14]、そこでの最適構造とは(正の重みに限定して考え)重みの総和が最小になる場合である。一般に、そのとき各入力変数の重みはしきい値関数であるための最小値になる。[13]では、2変数の最適構造のうち、重みが3になる場合(4以上はなし)が示されている。すなわち、3値2変数しきい値関数の「最大重み」は3である。

したがって3値2変数の場合、 $-2 \leq w_i \leq 2$ の整数ではすべてのしきい値関数を表し得ないから、重み w_i の初期値は、

$$w_i \in \{0, \pm 1, \pm 2, \pm 3\}$$

でなければならない。これらを一つの遺伝子型の重み部分にランダムに設定していく。このとき、 $W = (w_1, w_2)$ と入力 $X = (x_1, x_2)$ の内積 WX は最大18をとり得るので、しきい値 t_j は

$$t_j \in \{0, \pm 1, \pm 2, \dots, \pm 18\}$$

を満たすように設定しなければならない。

一般の p 値 n 変数($n \geq 2$)では式(11)のようにまとめられる。ここで $w_{\max}$ は最大重みである。

$$\begin{aligned} -w_{\max} &\leq \text{重み } w_i \leq w_{\max} \\ -(p-1)n \cdot w_{\max} &\leq \text{しきい値 } t_j \leq (p-1)n \cdot w_{\max} \end{aligned} \quad (11)$$

このことについて、次の簡単な補題が成り立つ。

[補題] 表5のように重みとしきい値の候補を並べて遺伝子型を定義する。こうして定めた遺伝子型を一つの行に並べ、 N 行(すなわち N 個)配置する。重みに対応する列の値としきい値に対応する列の値がそれぞれ式(11)を満たすとき、適当に $unit$ を定め N をいくらでも大きくできるならば、対象の回路網を実現する(教師信号と完全一致する)重みとしきい値を持つ遺伝子型が必ず存在する。

(証明) 公知のように多値しきい値関数は多値論理関数

に対して完全系である[14]から、対象の多値論理関数を実現する有限個のしきい値素子を用いる回路網が必ず存在する。このときの各しきい値素子の最適構造において重みとしきい値は式(11)を満たすので、 N を大きくとったとき、それらを含む遺伝子型が存在することは明らかである。□

4値以上の関数表は報告されておらず、また変数が増えると一般に最適構造における重みの最大値(最大重み)も増大するが、増加の仕方は2値しきい値関数の場合[14]から推測しても線形ではない。しかも、多値しきい値関数の最大重みについては2値の場合と異なりその大まかな上限値、下限値も知られていない。

重みの候補は、実験プログラム上で一様乱数を用いて設定するが、補題を満たすべく一つの列に式(11)を満たす値が設定されるようにする。以下の実験では、これら考察と乱数の精度に依存した解のばらつきを考慮し、次のように重みを表5形式の表中に初期値設定する。 p 値 n 変数しきい値関数の場合として述べる。

- 1) 最大重み $w_{\max}$ を仮定する。ただし、3値2変数、3変数については[13]の結果(3値2変数は3, 3変数は13)を用いる。
- 2) 乱数 $r \in \{0, \pm 1, \pm 2, \dots, \pm(w_{\max} + \alpha)\}$ を生成し、重み w_i とする。
- 3) $w_{\max} + \alpha$ を新たに $w_{\max}$ と考えて、式(11)を満たすように乱数を用いて各素子のしきい値 $t_1, t_2, \dots, t_{p-1}$ を定める。 $t_1 \leq t_2 \leq \dots \leq t_{p-1}$ を満たさない時は、並べ替えを行なう。

$w_{\max}$ をいくらに定めるかで実験結果が大きく左右されるが、これらについては後に述べる。一様乱数の発生にはばらつきがあり、少ない個数(N)の遺伝子型では $\pm w_{\max}$ までの整数が生成されない恐れがある。2) の α はそれを補完する目的で導入した余裕定数であり、実験結果からは2~4が適切のようである。

手順3) 適応度の算出：

遺伝子型を一つ取り上げ、適応度を算出する。ここで、この遺伝子型の重みとしきい値のもとで n 変数入力を与えたとき、素子 $u_1, u_2, \dots$ の出力を求める。入力の総個数を n_{total} とすると $n_{\text{total}} = p^n$ であるが、出力ユニットが n_{correct} 個の正しい値($f(x)$ と同じ値)を出力したとき

$$\text{適応度} = n_{\text{correct}} / n_{\text{total}}$$

で算出する。適応度=1ならばそのときの各素子の重みとしきい値を表示して終了する。このようにして、すべての遺伝子型について適応度をチェックする。すべての遺伝子型の適応度の平均値をこの世代の平均適応

度、適応度中の最大値を最大適応度という。

手順4) 交叉と関連処理：

現在の世代のすべての遺伝子型につき手順3)において最大適応度=1のものがいないときは交叉を行う。ここでは一様交叉[15]を実行する。このとき、親1または親2から子へランダムに受け継がれる遺伝子の確率は0.5とする。すべての遺伝子型をその適応度の大きい順に並び替え、平均適応度の低いものを淘汰(削除)し、一様交叉で得た子と入れ替える。淘汰確率は0.4に固定する。一様交叉の縁組みは淘汰の予定のない遺伝子型からランダムに選択する。また、交叉の際、2%程度の低い確率で遺伝子型の要素をランダムに入れ替え、突然変異を起こす。さらに、交叉の結果、各素子のしきい値の大小が $t_1 \leq t_2 \leq \dots \leq t_{p-1}$ を満たしているかどうか調べ、満たさない場合は並べ替えを行う。

以上の設定で QuickC によりプログラムを記述し、世代交代シミュレーションを実行した。なお、見出された適応度=1の回路網において常に0の信号を出している素子は、回路網の出力に影響していないので削除するルーチンを加えている。また、ある素子から出力素子への経路がない場合、重みがすべて0になることが考えられるが、表5の形式ではしきい値も記入しており、また多値しきい値関数のしきい値は複数なので表から見出すことは困難である。この点、2値の場合と異なる。

5. 実験結果とその評価

5.1 3.2で述べた問題点を含む推論規則表への適用

[19]で、3.2で述べた問題点を持つ二つの規則表に対して4の処理を行ない表の関数をしきい値素子回路網で実現できることを報告した。実現された表の一つは表4(a)の「unate だが非線形なファジィ推論規則表」で、他は表6の「unate でなくかつ非線形な規則表」である。前者は5値2変数の論理関数の実現問題であり、適応度1の回路網が173世代から得られた。後者は当初[3]で例示された時点では塗りつぶし字部分はなく、入力変数 x_1 は4値、 x_2 は3値、出力は3値をとる変則的なものであった。[19]で未定義の部分を don't care とみなして表6イタリック字の入出力値を追加定義し、4の処理を実行した。適応度1の回路網が326世代から得られた。[19]にこれら二つの実験における適応度の変化を示すグラフおよび得られた構造の例が示されているのでここでは再録しない。

本論文ではこれら二つの表の関数に対してさらに4.2のようにしきい論理の知見を応用して、重み、しき

い値の初期値設定を行なうよう修正し、ランダムな20個の乱数の seeds を与えてそれぞれ2000回の世代交代実験を行なった。また[19]での成功例に合わせて、表4(a)では素子の個数(*unit*)を11(実際は重み, しきい値がすべて0の行が現われることがあるので11以下である), 集団サイズを30に固定した。表6では素子の個数12(以下), 集団サイズを30に固定した。

この実験の目的は, 完全一致の回路網が得られることと $wmax$ の値が関係しているかを知ることである。結果を表7に示す。ここで, 塗りつぶしになっている「成功した試行数」とは20回の試行中で, 世代交代2000回までに完全一致の回路網が得られた試行数である。また「最短世代」とは, 成功した試行中, 見出された最初の世代数の内, 最小のもの, 「最遅世代」とは最大のものを指す。これら最初の世代数の平均値が「世代の平均値」である。

表8に, 表4(a)の5値関数に対する試行で, $wmax = 9.0$ のとき, 最短の実現世代が得られたときの重みとしきい値を示す。しきい値ユニットは u_1 から u_{11} まで使われているが, 表では個数が多いので添字のみ示す。他の試行の重み, しきい値は割愛する。

[実験結果の評価・考察]

わずかに二つの関数に対する実験であるが「成功した試行数」に傾向が見られる。表7だけから判断すると表4(a)の5値関数は $wmax = 9.0$, 表6の4値関数は $wmax = 7.0$ にピークがあるようであるが, このことは, 次のような説明が可能である。

「 $wmax$ のピーク値よりやや下(−1または−2)に理論的な最大重みがあると予想される。その値より $wmax$ が小さい試行では, 式(11)を満たすように与えられた重み・しきい値ですべての多値しきい値関数を表現できるとは限らないため, 成功回数が減少する。逆にその値より $wmax$ が大きい試行では, 重み・しきい値には大きな範囲の整数値が与えられるため線形加算値 WX としきい値の差が関数値の違いに対応しない場合が増え, 成功回数が減少する。」

最良の結果を与える $wmax$ のピーク値が, 予想される最大重みよりやや大きくなるのは, 乱数発生において偏りが生じ, $wmax$ までの値が必しも出現しない場合があるからと考えられる。

一つの $wmax$ に対する試行回数が少ないため, 最短世代, 最遅世代, 世代の平均値には顕著な傾向は見られないが, 表4(a)の5値関数では $wmax = 9.0$ のときすべての面で最良になっている。

表6 non-unate で非線形な規則表

$x_2 \backslash x_1$	0	1	2	3
0	0	1	1	2
1	1	2	2	1
2	2	1	1	0
3	1	0	0	0

表7 完全一致の回路網が得られた最初の世代

	表4(a)の5値関数			表6の4値関数		
$wmax$	8.0	9.0	10.0	6.0	7.0	8.0
試行回数	20	20	20	20	20	20
成功した試行数	12	15	12	6	12	5
最短世代	84	70	94	125	187	147
最遅世代	1474	1129	1140	914	1108	1794
世代の平均値	382	315	440	347	518	728

(世代の平均値は小数点以下四捨五入)

表8 表4(a)の5値関数を実現する最短世代の重みとしきい値

to	from													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	5	12	−1	5	22	0	0	0	0	0	0	0	0
4	0	3	−7	−25	4	4	22	0	0	0	0	0	0	0
5	3	−2	0	−2	8	−8	7	20	0	0	0	0	0	0
6	4	8	0	8	−6	−14	−8	8	9	0	0	0	0	0
7	5	6	−8	−1	−7	−1	−9	2	8	12	0	0	0	0
8	9	9	1	3	6	0	−6	−13	−11	−3	−3	0	0	0
9	6	8	1	5	3	0	4	9	−17	−10	3	5	0	0
10	7	0	−4	−3	−3	6	5	4	1	−13	−7	8	15	0
11	7	7	1	6	−5	2	9	−5	−1	−5	−23	−7	9	26

(第70世代, 塗りつぶしはしきい値)

いずれにしても, 一般的な結論を導くにはさらに大量の実験が必要と考えるが, 4.2のようにしきい論理の知見を利用することは一つの有効な方策であるとみてよいと思われる。

なお4.2で少し触れたが, 多値 n 変数しきい値関数の最大重みを求めることはそれ自体大きな問題であって, 完結した式で表すことはもとより大まかな上限値, 下限値さえ明らかになっていない。

5.2 ここでの手法の能力と限界

本論文で述べた手法の能力をみるために, サイクリック演算をとり上げる。表9の3値論理関数はサイクリック演算をすべて含んでおり, 線形性, unate 性などしきい値関数で実現し易い性質から最も遠い関数の一つと考えられる。

表9 サイクリック演算を含む
3値論理関数

x_1	x_2		
	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

5.1と同様に、そこでの乱数列を用い $wmax$ を入れ替え、各試行2000回までの世代交代によりしきい値素子回路網実現を試みた結果を表10に示す。ここで、素子数は11、集団サイズは30に固定している。

[実験結果の評価・考察]

すでに知られている最大重み3のとき[13]ではなく、 $wmax=5.0$ のとき最良の結果が得られている。世代の平均値は2000回までの世代交代で求める回路網が見出された場合の平均なので成功した試行数が小さいと意味がない。

問題は成功しなかった試行でさらに2000回を超える世代交代を行なったとき、必ず適合度1の回路網が見出されるかどうかである。詳しくは述べないがこの例では完全一致解が得られたケースと、適合度の極大値に陥って循環しているとみられるケースに分かれた。しかし後者の場合でも異なる乱数列を与えると早い世代で求める解が得られた。これらは、結果の成否が初期値設定の偶然性にかなり依存していることを示しているといえよう。図5に $wmax=5.0$ の最短世代が見出されたときの平均適応度、最大適応度のグラフ、表11に得られた最短世代115のときの重みとしきい値を示す。

図5 表9の関数の最短世代における最大適応度
(実線)と平均適応度(点)の変化

表10 表9の関数の実現実験

$wmax$	3.0	4.0	5.0	6.0
試行回数	20	20	20	20
成功した試行数	2	9	11	9
最短世代	167	148	115	137
最遅世代	265	1650	1381	1313
世代の平均値	216	632	513	571

次に、ランダムに生成した表12で1)から10)の10個の3値2変数論理関数 f に対して5.1の手法を実行した結果を示す。ここでは、 $wmax=5.0$ であるが、素子数、集団サイズをそれぞれ13、50と大きくとり、一つの乱数列で実現できなかった場合他の乱数列で実行したため、すべて回路網が得られている。

[3変数以上の関数実現における問題点]

最後に、3変数以上の関数の実現について述べる。3変数以上の多値論理関数への手法の適用は教師信号と

表11 図5で得られた最短世代の構造

to	from											
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	11	12
u_1	0	0	0	0	0	0	0	0	0	0	0	0
u_2	0	0	0	0	0	0	0	0	0	0	0	0
u_3	-2	-2	-6	-4	0	0	0	0	0	0	0	0
u_4	5	5	3	0	1	0	0	0	0	0	0	0
u_5	3	2	-3	-1	0	2	0	0	0	0	0	0
u_6	-5	-1	3	-4	-2	0	0	0	0	0	0	0
u_7	-2	-4	2	-1	0	3	-6	9	0	0	0	0
u_8	1	-3	0	2	5	-2	5	-6	13	0	0	0
u_9	-5	-2	-2	0	0	-5	-4	0	-16	4	0	0
u_{10}	0	2	2	0	0	-4	-1	0	-5	-8	-4	0
u_{11}	3	3	-3	3	-5	-4	5	-2	0	-4	-4	-2

(第115世代、塗りつぶしはしきい値)

表12 ランダムな3値論理関数の実現

f	x_1	0	0	0	1	1	1	2	2	2	最初の 実現世代
	x_2	0	1	2	0	1	2	0	1	2	
1)	0	0	0	1	2	2	2	1	2		166
2)	2	2	1	0	0	1	0	1	2		249
3)	0	2	2	0	2	1	2	1	2		1209
4)	0	1	0	2	0	2	0	2	0		646
5)	1	0	2	2	2	0	1	1	0		213
6)	2	2	1	1	2	2	0	1	1		267
7)	0	2	0	2	2	1	2	1	1		1394
8)	1	2	2	1	1	1	2	1	2		89
9)	2	1	2	1	2	2	0	2	1		299
10)	1	0	1	1	2	2	2	2	2		57

の照合箇所が増えるため急激に能力が低下する。5.1のようにしきい論理の知見を利用しても、遺伝的アルゴリズムでいわれている短所：「最適解の近傍までは非常に速く収束することができるが、そこから最適解への探索が苦手[17]」が現われ、完全一致までには多くの世代交代が必要となる。GAにおける積み木仮説[17]としきい値関数の完全性によれば、いくらでも多くの遺伝子型と素子を使い、重み・しきい値の初期値設定で十分な値が与えられていればここでの手法で理論的には完全一致解が得られる筈である。しかし、実行時間、必要なメモリ数等から考えてこのままでは実用的でない。単純GAに他の手法をさらに組み合わせしていく必要がある。

しかしながら、本論文の手法の本来のきっかけである「推論規則」においては、一般にunate性や線形性以外にも一定の法則性(たとえば、きちんと定義されてはいないが変化が急激でないことなど)がみられる。本論文の手法でそれを表現することは3変数以上でも困難ではないと考えられ、こうした観点からの関数の複雑さについてきちんとした定義・考察が必要になったと考えている。

6. むすび

ファジィ理論にGAを応用する研究は、メンバーシップ関数の同定・チューニングにGAを用いるケースがあるが他の応用は見られないようである。一方、ニューラルネットをファジィに應用するケースとしてKosko[11]が知られているが、これもメンバーシップ関数のチューニングにBP法を用いるもので、その後の処理は従来のファジィ推論法を用いている。

推論規則表は一つのまとまった知識である。このまとまった推論知識を一つのニューラルネットで表現しようとするのが本論文のゴールだといえよう。そこには本来のファジィ集合・メンバーシップ関数の考え方は不要である。すなわち、推論規則のラベルを連続する整数値で置き換えて得た関数を多値論理関数とみてGAを利用したニューラルネット実現を図る。実用には、得られた回路網のしきい値素子を[1]の無限多値しきい値素子(実数入出力)で置き換え、さらに入出力ゲインを調整することにより、実数全体をカバーする推論エンジンとして使うことができる。

GAとニューラルネットとの融合は、GAによるニューラルネットのトレーニングや構造設計など多くの研究がある[17]。本論文は、用いたGAの手法(交叉など)に目新しいものはないが、多値論理関数に対応する格子点での完全一致を目指すためにしきい論理の知見を応用している。しかしながら、多変数になると照合

箇所が急激に増えるため、解の近傍までは達するものの実用的には完全一致が困難になる。加えて、結果の成否が重み、しきい値の初期値設定にかなり依存していることは明らかで、これらの数値を自己学習させる仕組みを取り入れるなど単純GAに他の手法を組み合わせしていく必要がある。これらを解決しない限り、「完全一致解を見つけることが本来不得意[17]と言われるGAをなぜ用いるのか」という実は根本的な問題が発生してくる。

なお、前述した規則表の回路網実現は、最終的には実数重みのニューラルネットを構成することであるので、Montana and Davis[16]の研究が関わりを持つと予想される。しかし彼らの実験は誤差の最小二乗和でよしとするもので、たとえ遺伝子型、素子および世代交代を増やしても完全一致に至る理論的な根拠は説明されていないことを指摘しておきたい。

参 考 文 献

- [1]山本喜則：“ファジィしきい値関数の定義とその応用”，日本ファジィ学会誌，Vol.4，No.2，pp.303-315，1992.
- [2]Yoshinori Yamamoto：“A Meaningful infinite-valued switching function,” Proc. IEEE Int. Symp. on Multiple-valued Logic, pp.415-422, May1992.
- [3]山本喜則：“近似推論へ応用可能な無限多値多重しきい値関数”，日本ファジィ学会誌，Vol.6，No.1，pp.141-151，1994.
- [4]Saburo Muroga, Threshold logic and its applications, Wiley & Sons, 1971.
- [5]S.T.Hu, Threshold logic, University of California Press, 1965.
- [6]三根，藤田：“多値論理関数の分解とそのしきい値回路網への応用”，電子通信学会論文誌，Vol.56-D，No.12，pp.681-688，1973.
- [7]J.Santos and F.Lorenzo：“Threshold synthesis for ternary digital systems,” IEEE Trans, Vol.EC-15, No.2, 1966.
- [8]水本雅晴：ファジィ理論とその応用，サイエンス社，pp.146-149，1990.
- [9]麻生英樹，ニューラルネット情報処理，産業図書，p.17，1988.
- [10]山本喜則：“無限値しきい値関数の数学的性質と多段結合”，日本ファジィ学会誌，Vol.9，No.4，pp.560-569，1997.
- [11]Bart Kosko：Neural networks and fuzzy systems, Prentice Hall, pp.345-361, 1992.
- [12]三根，藤田：“三値しきい値関数における整数分離系”，電子通信学会論文誌，Vol.53-C，No.12，pp.970-971，1970.
- [13]相原，赤城：“三変数までの三値しきい値関数の作成”，電子通信学会論文誌，Vol.53-C，No.9，pp.591-598，1970.
- [14]室賀，茨木，北橋，しきい論理，産業図書，pp.153-

- 172, 1976.
- [15] Nagao, Agui and Nagahashi : "Structural evolution of neural networks having arbitrary connections by a Genetic method," IEICE Transaction on Information and Systems, Vol.E76-D, No.6, pp.689-697, 1993.
- [16] D.J.Montana and L.D.Davis : "Training feedforward networks using genetic algorithms," Proc. of the Int. Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1989.
- [17] 北野宏明, 遺伝的アルゴリズム, 産業図書, pp.235-262, 1995.
- [18] 山本, 藤田 : "ファジー多数決関数", 電子情報通信学会論文誌, Vol.J73-D-I, No.8, pp.673-682, 1990.

- [19] Yoshinori Yamamoto : "A synthesis method of the approximate reasoning engine by means of Genetic Algorithm," Proc. IEEE Int. Symp. on Multiple-valued Logic, pp.201-208, May 1998.

(1998年10月13日 受付)

[問い合わせ先]

〒370-0801 高崎市上並榎町1300

高崎経済大学経営学科

山本 喜則

TEL : 027-344-7520

E-mail : yamamo@namesv01.tcue.ac.jp

—— 著 者 紹 介 ——

山本 喜則(やまもと よしのり)

高崎経済大学経済学部経営学科

昭和46年3月 東京教育大学理学部応用数理学科卒業。昭和57年 京都大学工学博士。岡山理科大学助手、高崎経済大学講師、助教授等を経て高崎経済大学教授。多値論理における特殊関数(多数決関数、しきい値関数、誤り訂正可能な論理関数)、多値論理回路の故障検出、多値論理応用近似推論法等の研究に従事。社会科学分野ではテレワーク、情報機器のリサイクル問題などを手がけている。電子情報通信学会、IEEE、日本ファジィ学会、日本テレワーク学会会員。

A Synthesis of an Infinitely Multi-valued Approximate Reasoning Engine by Means of Genetic Algorithm

by

Yoshinori YAMAMOTO

Abstract :

In the past papers, the author proposed a synthesis method of the approximate reasoning engine by means of a newly defined functions : the infinitely multi-valued threshold functions. The essential point of the method is to represent the hypersurface which a rule table expresses by the formula of the infinitely multi-valued threshold function. It is shown in the papers that the proposed method is effective to the process control. This paper discusses an unsolved problem in the synthesis method. The problem is to represent the rule table having the complicated properties such as non-unateness/non-linearity etc. by one formula. As shown in the previous paper[10], a targetting hypersurface may be approximately realized by obtaining the finitely multi-valued logic function which the lattice points(corresponding to integer element I/Os) express and by giving analogue inputs to the function's formula. Thus, the question results in the realization problem of any finitely multi-valued logic function by threshold gates. For the problem, however, no computer-oriented simple methods were known. This paper proposes a multistage synthesis method of the threshold gates by using Genetic Algorithm and Threshold Theory. The point of the method is as follows. First, we consider genotypes built of integer elements. We set initial integer values in the genotypes so that "the maximum weight" of a finitely multi-valued threshold function may be generated. Next, we apply the genetic algorithm to the genotypes. The complicated rule tables shown in the previous papers have been realized by a network of the threshold gates assuming analogue inputs. Together with the experiments of the realization, the capability of the method is discussed through another examples.

Contact Address : Yoshinori YAMAMOTO

Dept. of Business Administration, Takasaki City University of Economics

1300, Kaminamie-Machi, Takasaki-Shi, 370-0801, Japan

TEL : 027-344-7520

E-mail : yamamo@namesv01.tcue.ac.jp