

琉球大学学術リポジトリ

遺伝子長可変型GAによる4値SVTゲート回路網の合成

メタデータ	言語: 出版者: 琉球大学工学部 公開日: 2009-12-18 キーワード (Ja): 遺伝子長可変型GA, 可変しきい値ゲート, 突然変異 キーワード (En): 作成者: 瑞慶覧, 長定, 島袋, 勝彦, 新垣, 栄, 小波津, 清正, Zukeran, Chotei, Shimabukuro, Katsuhiko, Arakaki, Sakae, Kohatsu, Kiyomasa メールアドレス: 所属:
URL	http://hdl.handle.net/20.500.12000/14149

遺伝子長可変型GAによる4値SVTゲート回路網の合成 Synthesis of Quaternary SVT-gate Networks using Genetic Algorithm with Variable Gene Length

瑞慶覧 長定* 島袋 勝彦* 新垣 栄** 小波津 清正***
Chotei ZUKERAN* Katsuhiko SHIMABUKURO* Sakae ARAKAKI** Kiyomasa KOHATSU***
琉球大学工学部
Faculty of Engineering, University of the Ryukyus

Abstract : We have confirmed that genetic algorithm(GA) with variable gene length is a useful method compared to GA with fixed one for synthesizing SVT-gate networks of quaternary logic function. However, it is difficult to obtain the optimal solution of given function, because there are many choices of possible threshold functions during the synthesis. In this paper, we introduce a new mutation process for improving solution, and discuss the ability of the GA for searching optimal solution of the synthesizing SVT-gate networks for quaternary logic function.

キーワード : 遺伝子長可変型GA、可変しきい値ゲート、突然変異

1. はじめに

4値論理系における基本演算子であるSVTゲートは、しきい値制御端子を有し可変しきい値ゲートと呼ばれる。任意の4値n変数関数は4ⁿ-1個のSVTゲートを用いて、樹枝状構造で標準展開できる。

SVTゲート回路網においては、ゲートのしきい値制御関数の決定の仕方により、ゲート数の大幅な減少が可能であるが、それらしきい値特性を活かした合成法は、しきい値関数の多様性から、最適解の導出が非常に困難である。

この問題解決のために、最適化の一手法である遺伝的アルゴリズム(GA) [1][2]を応用する。4値SVTゲート回路網をコーディングした遺伝子長固定型GAを用いた合成法は、ある程度良好な結果を得ることができた。[3]しかしながら、このコーディング方法における4値SVTゲート回路網は、標準展開型をベースとしているため、表現できる回路網に制限があり、変数が増加する度にベースとなる回路網と、染色体のコーディングを作成し直さなければならない。また、変数が増加するに伴い、標準展開型回路網に使用するゲート数が増加するため、遺伝子長の増加、探索空間の拡大につながる。

そこでより自由で柔軟な回路網表現を行なうため、遺伝的アルゴリズムを構造的表現ができるように拡張し、遺伝子長を可変とした染色体を用いるGAを提案し、良好な回路網合成が行なえることを示す。

2. 4値SVTゲート回路網の合成

4値論理系における論理値の集合をL={0,1,2,3}とすれば、4値SVTゲートの出力S(a,b;i;c)は式(1)の様に定義される。[4][5]

$$S(a, b; i; c) = \begin{cases} a & : i \leq c \\ b & : \text{otherwise} \end{cases} \quad (1)$$

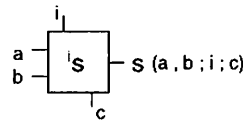


図 2.1 SVTゲートのシンボル

表 2.1 真理値表

i \ c	0	1	2	3
0	a	a	a	a
1	b	a	a	a
2	b	b	a	a
3	b	b	b	a

但し、4値定数 a,b,i;c ∈ L である。

ここで、iをしきい値制御変数、cを制御変数、a,bをそれぞれ Residue function a,bと呼ぶ。以後 Residue function a,bをそれぞれ Res_a, Res_bと略する。

式(1)に従い、4値SVTゲート回路網の単純化に利用できるSVTゲートの性質を以下に示す。

$$S(f(x'); f(x'); i; c) = f(x') \quad (2)$$

$$S(a, b; x; 3) = S(a, b; 0; x) = a \quad (3)$$

但し、f(x')は任意の4値論理関数とし、a,b,i,c ∈ {0,1,2,3}とする。

式(2)と(3)の左辺は、回路網合成には冗長な表現を表しているが、右辺の表現に変換することにより冗長なゲートを省くことができ、単純化に利用される。

3. SVTゲート回路網と遺伝子長可変型GA

3.1 遺伝子長可変型GA [6]

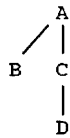
一般的に用いられてる遺伝子長固定型GAでは、対象となる問題の最適解が存在すると思われる範囲の解候補全てを、個体として表現できる遺伝子型にコーディングを行ない、最適解の探索を行なっている。

GAを用いて4値2変数関数の合成を行なうには、回路網を染色体で表現しなければならない。しかし、遺伝子長を固定すると、表現できる回路網が限定されてしまう。

受理 : 1997年5月26日
*工学部電気電子工学科
(Dept. of Electrical and Electronic Engineering, Fac. of Eng.)
**大学院工学研究科電気電子工学専攻
(Div. of Graduate School, Electrical and Electronic Eng.)
***沖縄電力株式会社
(Okinawa Electric Power Company, Inc.)
日本ファジィ学会九州支部7月研究例会において1996年7月に発表済み

また前述した様に、最適なしきい値関数の決定が非常に困難なため、しきい値関数を生成するしきい値回路網の設定も難しいものとなっており、できるだけ多くのしきい値関数を生成できるようにしきい値回路網を設定すると、想定する回路網全体が大きくなってしまい、遺伝子長の増加、探索空間の拡大につながってしまう。

本研究で提案する遺伝子長可変型GAでは、遺伝子長の異なる染色体を用いる事で、自由なしきい値関数を生成できるようになり、しきい値回路網の決定が容易になる。遺伝子長可変型GAは、従来のGAを拡張し構造的な表現を扱えるようにしたものであり、ここでは木と呼ばれる構造表現を扱う。木とはサイクルを持たないグラフのことであり、次のような構造をいう。



このような木構造に関して、以下の用語を用いる。

- ノード：記号A, B, C, Dのこと
- 根(ルート)：A
- 終端ノード：B, D
- 非終端ノード：A, C

遺伝子長可変型GAでは、1個のSVTゲート $S(a,b;i;c)$ をリスト $[i,c,a,b]$ で表し、これを表現型と呼ぶ。また、個体(SVTゲート回路網)を数値系列で表現したものを遺伝子型といい、表現型と遺伝子型との対応を以下の表の様に定義する。

表3.1 表現型と遺伝子型のコーディング法

表現型	0	1	2	3	x	y	[]
遺伝子型	0	1	2	3	4	5	6	7

0~3：4値定数
 x, y：入力変数
 [,]：ゲート区切りとする。

表3.1の対応表をもとに、図3.1(a)(b)の回路網を表現型から遺伝子型へ変換すると以下の様になる。

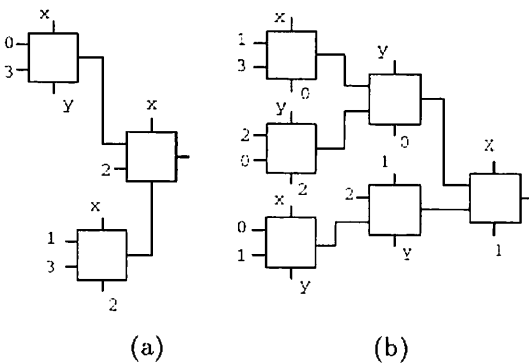


図3.1

(a): $[x,[x,2,1,3],[x,y,0,3],2]$
 (b): $[x,1,[y,0,[x,0,1,3],[y,2,2,0]],[1,y,2,[x,y,0,1]]]$



(a):6464213764503727
 (b):6416506401376522077615264501777

この様に、回路網で使用されているSVTゲート数の違いにより、遺伝子長が異なってくるのである。

ここで、 $S(a,b;i;c)$ のリスト表現 $[i,c,a,b]$ において、各変数の順序が違うのは、プログラムの都合のためであり、特別な意味はない。

3.2 SVTゲート回路網合成アルゴリズム

遺伝子長可変型GAによるアルゴリズムをフローチャートにすると以下の様になる。

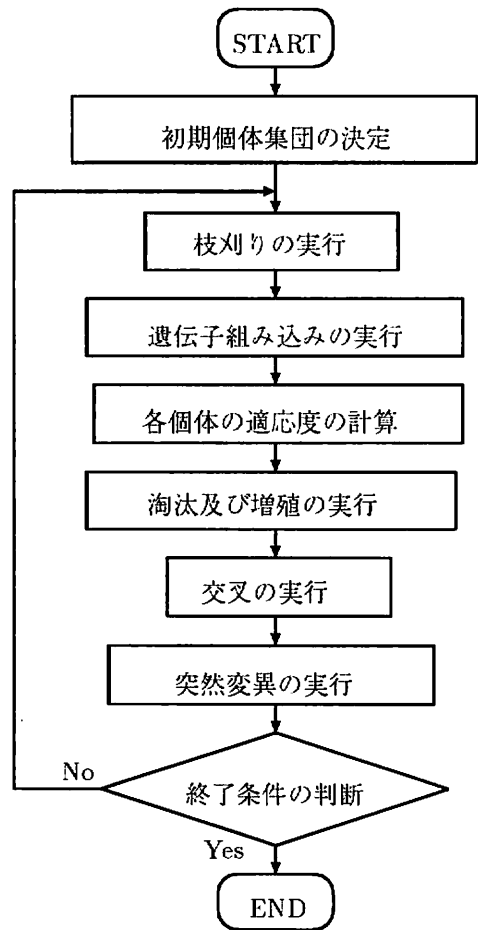


図3.2 フローチャート

ここで初期個体集団の決定は、乱数を用いて行なうため、ゲート未使用の個体や、3個のSVTゲートを使用している個体など、遺伝子長の異なる様々な個体が生成される。

また、終了条件の判断として本研究では、世代交代回数があらかじめ定めた回数に達した時を終了条件とした。次に、各部の説明を行なう。

3.2.1 適応度の計算

各個体の適応度の計算方法を以下の様に定義する.

- 関数合成ができていない場合、与えられた4値2変数関数 $f(x,y)$ の真理値表の中で、正しく出力されたセルの数を個体の適応度とする.
- $f(x,y)$ の真理値表の関数値を全て正しく出力した場合、その回路網で使用されているSVTゲートの数と、標準展開型回路網で使用されているSVTゲートの個数15 ($= 4^2 - 1$)との差をとり、その値に $f(x,y)$ の真理値表のセル数16を加えたものを個体の適応度とする.

つまり、問題として与えられた4値2変数関数が合成可能で、使用しているSVTゲート数が少ない回路網を表現できる個体ほど、高い適応度が与えられている.

3.2.2 交叉について

2つの隣合った個体間で遺伝子を組み替えて新しい個体を生成するプロセスでは、交差位置はランダムに決定される. しかし、本研究で提案する遺伝子長可変型GAでは同一世代における各個体の遺伝子長が異なるため、従来の交叉方法を用いることができない. そこで、各個体の表現する樹枝状回路網である木構造を意識し、木構造における終端ノード、つまりSVTゲートの各端子(しきい値制御・制御・Res_a・Res_b)、部分回路網及び回路網全体のそれぞれの間で行なう.

ここでの交叉は、以下のケースから選ばれる.

case 1 部分回路網の入れ換え

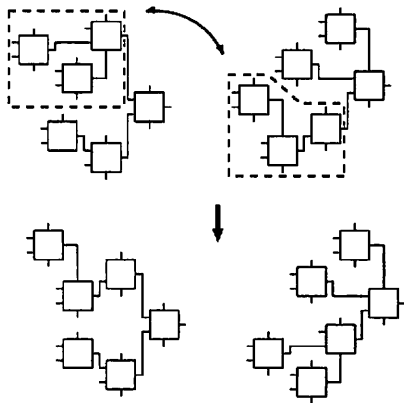


図 3.3

case 2 部分回路網と回路網全体の入れ換え

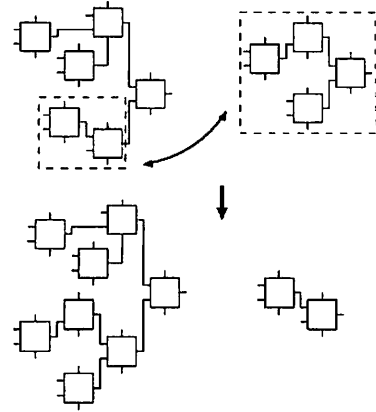


図 3.4

case 3 部分回路網と1端子の入れ換え
($a \in \{0, 1, 2, 3, x, y\}$)

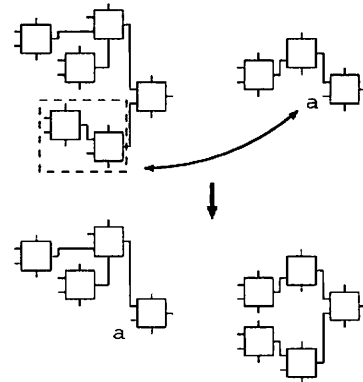


図 3.5

case 4 回路網全体と1端子の入れ換え
($a \in \{0, 1, 2, 3, x, y\}$)

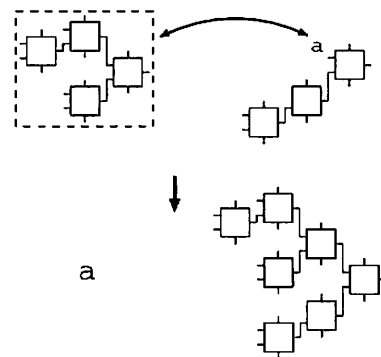


図 3.6

また、交叉によって回路網の形状もしくは使用しているゲート数に変化を与えるため、次のような制限を行なう.

case 5 ゲートを1個だけ使用している個体同士の交叉においては、1端子同士の入れ替えを行っても、各個体が表現する回路網に変化が少ないため、必ずどちらかのゲートがもう一方の回路網に組み込まれるように、交叉点を決定する。

$$(a \in \{0, 1, 2, 3, x, y\})$$

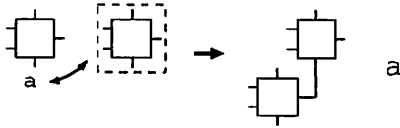


図 3. 7

case 6 ゲート未使用の個体との交差の場合、交叉により入れ替えが表現できるのは、ペアとなっている個体が表現するSVTゲート回路網の各端子（終端ノード）以外にない。case 5で述べた様に1端子同士の交叉は行わない事から、ゲートを2個以上使用している個体とゲート未使用の個体の組み合わせの場合には、ゲートを2個以上使用している個体の部分回路網と、ゲート未使用の個体を入れ替えるように、交叉点を限定する。

$$(a \in \{0, 1, 2, 3, x, y\})$$

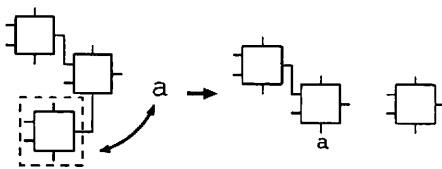


図 3. 8

更に、個体の組み合わせが以下のような場合には、生成される個体に変化がほとんどないため交叉を行わない。

case 7 双方ともゲート未使用

case 8 ゲートを1個だけ使用している個体とゲート未使用の個体

case 7の場合、交叉を行なう事は個体を並べ替えることであり、case 8の場合はゲート1個だけ使用している個体の1端子入れ替えになってしまうため、交叉を行わない。

3. 2. 3 突然変異

遺伝子のある部分の値を乱数により強制的に変える処理である。3. 1節で提案したコーディング表においてゲートの区切りを表す遺伝子があり、この遺伝子を突然変異により別の値に書き換えることは、ゲート1個もしくは部分回路網を回路網中から消去することになる。そのため、ここでは以下の様な突然変異を行う。

case 1 終端ノードから終端ノードへの突然変異

つまり突然変異は、終端ノードとみなせるSVTゲートの各端子に限定する。

しかし、以上のような処理だけでも回路網解の導出はできるものの、幾つかの関数は最適解の導出には至っていない。それは、前述した様に最適なしきい値関数の決定が困難なためである。

そこで、更に以下の様な突然変異を導入する。

case 2 終端ノードから非終端ノードへの突然変異

case 3 非終端ノードから終端ノードへの突然変異

case2 と case3 の処理は、SVTゲートのしきい値特性を活かした合成法を行うためのものである。しかしながら、全てのSVTゲートのしきい値制御・制御端子についてこの処理を行なうと、処理時間が増大するため、ここでは出力側から数えて1段目のゲートのしきい値制御・制御端子に対してのみ行う。これは、最適解の導出のために、1段目のゲートのしきい値制御・制御端子の組み合わせがより重要な役割を果たすと考えられるためである。

SVTゲート回路網における上記の3種類の突然変異の例を図3. 9～3. 11に示す。

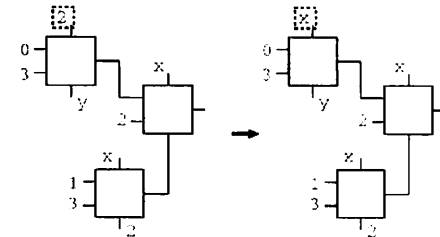


図 3. 9 突然変異 (case 1)

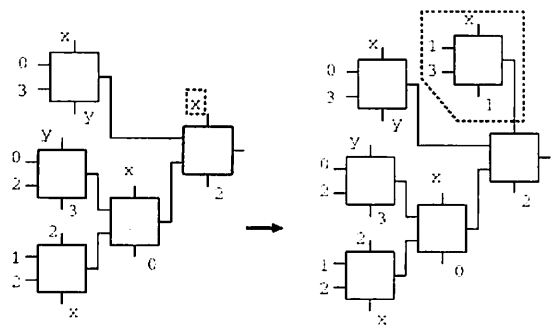


図 3. 10 突然変異 (case 2)

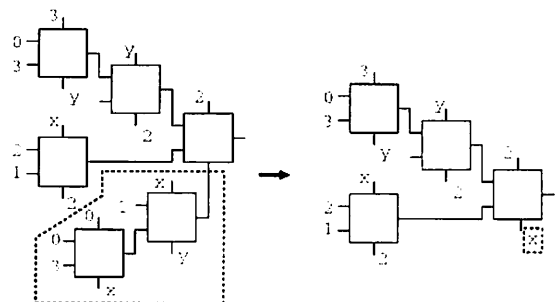


図 3. 11 突然変異 (case 3)

3.2.4 枝刈り

SVTゲートは(2)式に示される様な性質を持ち、しきい値制御・制御端子値により、Res_aもしくはRes_b値を出力する。しかし、(2)式のようなゲートや、Res_aもしくはRes_bのみを出力するしきい値制御・制御端子の組み合わせを持つゲート(例： $\{i,c\}=\{0,0\},\{2,1\},\{0,x\},\{y,y\}$ など)は回路網において何ら意味を持たず、無駄にゲート数を増やすだけである。

また、上記以外のしきい値制御・制御端子の組み合わせを持つゲートであっても、Don't careを含む部分関数を分割した際、生成されるRes_a,Res_b関数のどちらかが、Don't careのみの関数となってしまう様なしきい値制御・制御端子値の組み合わせを持つゲートは、関数を2分割することで合成を進めるSVTゲート回路網中において、無駄なゲートとなる。

このような条件を備えたゲートは、無駄なゲート数の増加・回路網拡大の原因であり、計算量の増大につながるため、回路網中から削除する。

また、しきい値制御関数を生成する部分回路網は、あまり複雑なものでもなくとも、十分に回路網全体の単純化に貢献できると考えられるため、しきい値側の部分回路網で使用されるゲート数のある範囲に限定し、余分なゲートを削除する。

以後、これらの不必要と考えられるゲート・部分回路網を個体中から削除する事を枝刈りと呼ぶ。

例)

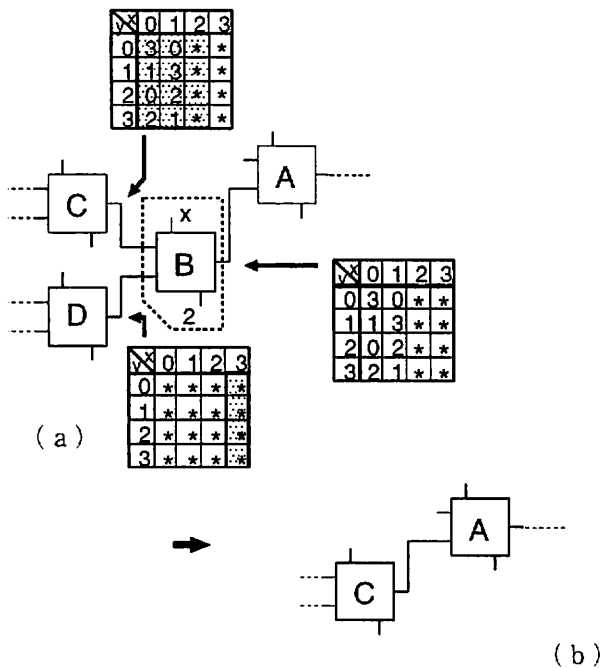


図3.12 Res_a,Res_b どちらか一方のみを出力する場合

図3.12の場合、生成されたゲートBのRes_b関数が全てDon't careの関数となっている。よって図(a)は、ゲートB及びそのRes_b側に続くゲートが消去され、図(b)と等価になる。

3.2.5 遺伝子組み込み

各個体の表現する回路網は、交叉により他の個体の部分回路網を取り込む事や突然変異を行なうことにより、その回路網を拡大する事ができる。しかし、取り込んだ部分回路網が関数合成について意味のあるものでなければ、枝刈りによって削除されてしまう。

より多くのゲートを必要とする関数を問題として与えた場合、回路網が拡大しなければ合成は不可能となる。合成できないというのは、その回路網で問題として与えられた関数を2分割していった際、生成されるResidue関数の内、4値定数もしくは入力変数とみなすことのできない関数が存在する、という事である。

そのような場合、乱数で生成したゲートを入力端子(Residue端子)に組み込み、上述の様なResidue関数を更に2分割し、関数合成を進める事を目的とした処理を行なう。以後、この処理を遺伝子組み込みと呼ぶ。

4. シミュレーション結果

本研究では、3.6で示した従来のcase1の突然変異のみの場合と、新しく提案したcase2,3の突然変異を導入した場合との比較を行なう。前者を”突然変異1”、後者を”突然変異2”とし、表4.1、4.2の4値2変数関数を与えた場合のそれぞれのシミュレーション結果を示す。

表4.1 4値2変数関数 $f_1(x,y)$

y \ x	0	1	2	3
0	0	1	0	1
1	2	3	2	1
2	0	2	3	2
3	3	0	0	1

表4.2 4値2変数関数 $f_2(x,y)$

y \ x	0	1	2	3
0	0	1	0	2
1	1	3	2	1
2	3	3	2	3
3	0	2	0	2

なお、今回のシミュレーションで用いたパラメータは以下の通りである。

- 選択淘汰 ... ルーレット戦略 + エリート戦略
- 世代交代数 ... 300
- 交叉率 ... 0.4
- 突然変異率
 - 突然変異2なし
 - 突然変異率1 ... 0.02
 - 突然変異2あり
 - 突然変異率1 ... 0.02
 - 突然変異率2 ... 0.1

関数 $f_1(x,y), f_2(x,y)$ について、回路網シミュレーションを行なった際の最大適応度と平均適応度の推移（5回平均）をそれぞれ図4.1、図4.2に示す。

突然変異2なし... { 最大適応度： ×
平均適応度： □
突然変異2あり... { 最大適応度： ⊖
平均適応度： △

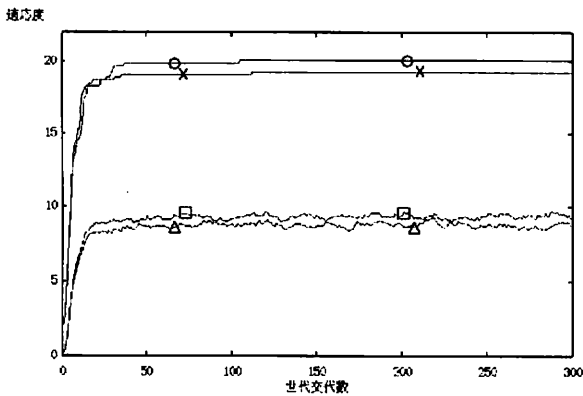


図4.1 関数 $f_1(x,y)$ についての最大適応度と平均適応度の推移

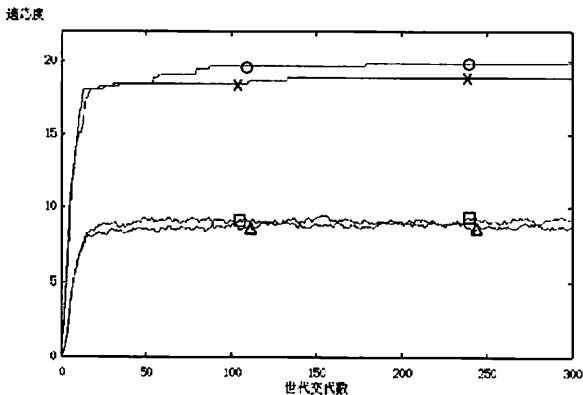


図4.2 関数 $f_2(x,y)$ についての最大適応度と平均適応度の推移

図4.1と図4.2より、突然変異2がある場合とない場合とでは、最大適応度の上昇については大きな差はないが、最終的な最大適応度の値は、突然変異2を導入した方が、より高くなっている。

次に、関数 $f_1(x,y), f_2(x,y)$ についての合成結果（5回ずつ試行させた中で、一番ゲート数が少ない解）をそれぞれ表4.3、表4.4に示す。

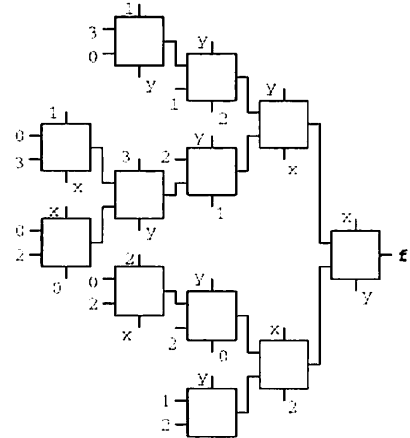
表4.3 関数 $f_1(x,y)$ の合成結果

	総ゲート数	最大適応度に達した世代数
突然変異2なし	12	19
突然変異2あり	11	23

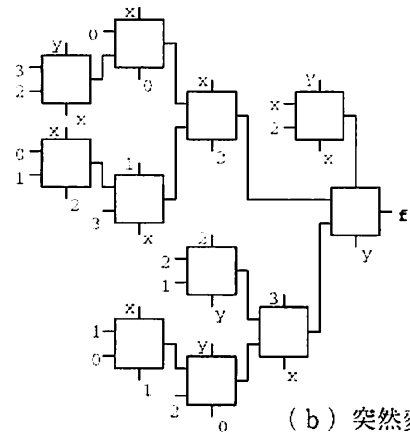
表4.4 関数 $f_2(x,y)$ の合成結果

	総ゲート数	最大適応度に達した世代数
突然変異2なし	12	110
突然変異2あり	10	179

また、上の結果で得られた回路網解をそれぞれ図4.3、図4.4に示す。



(a) 突然変異2なし



(b) 突然変異2あり

図4.3 関数 $f_1(x,y)$ のSVTゲート回路網解

表4.3、表4.4より、新しい突然変異を導入した場合は、ないときのそれと比べ、使用したSVTゲート数が少ないことがわかる。これは、回路図4.3、4.4を見てもわかるように、突然変異2を使用した方の回路の1段目の分割ゲートのしきい値側にゲートが接続されているためであり、そのため、関数の2分割がうまく行なわれ、ゲート数が減少したのである。

また図4.1、4.2において、新しい突然変異を導入したのにも関わらず、最大適応度の上昇は、ほぼ一致していることがわかる。

これらのことより、1段目のSVTゲートのしきい値制御・制御端子に集中した処理を行なうという新しい突然変異を導入することによって、最大適応度や平均適応度の成長を妨げることなく、よりSVTゲートのしきい値特性を活かした合成ができるようになった。

次に、乱数で生成した4値2変数関数2400個を新突然変異を使用しない場合と、使用した場合との2種類で試行させた結果（同一関数に対し5回ずつ試行させた中で、ゲート数が少ない解を選ぶ）を表4.5、図4.4に示す。

表4.5 回路網の分布

個数(個)	6	7	8	9	10
突然変異2なし	24	96	336	528	528
突然変異2あり	24	96	336	816	600

11	12	13	14
504	288	48	24
360	168	0	0

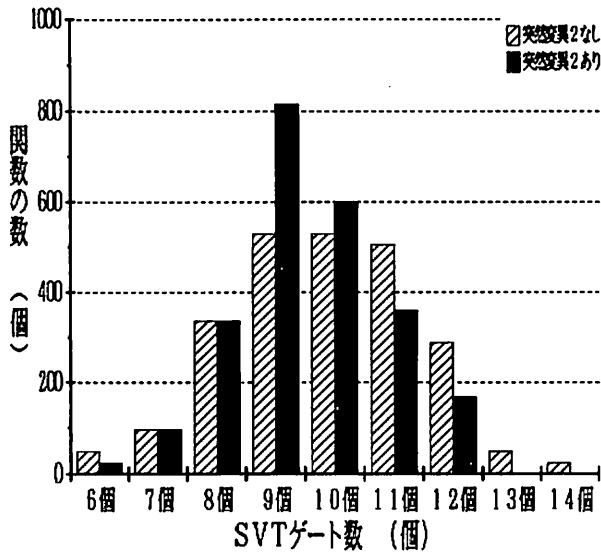


図4.4 回路網の分布

また、生成した2400個の回路網の平均SVTゲート数を表4.6に示す。

表4.6 平均SVTゲート数(関数2400個)

	平均SVTゲート数(個)
突然変異2なし	9.85
突然変異2あり	9.51

表4.6より、突然変異1のみの場合は平均SVTゲート数の減少が9.85に対し、新しい突然変異を導入した場合は平均SVTゲート数が9.51個という良好な結果が得られた。

また、図4.5より、ゲート数11個~12個の範囲の関数の数が大幅に減少していることがわかる。今回行ったシミュレーション結果において、新しい突然変異を導入した場合、乱数で生成された関数2400個全て、12個以下で合成できている。

これは、突然変異2という新しい処理を使用することにより、突然変異1のみの場合に比べ、より複雑なしきい値関数が生成されるため、ゲート数の減少につながったと考えられる。

以上のことより、新しい突然変異を導入することによって、ゲート数の減少が可能となった。しかしいくつかの関数に対しては、最適なしきい値関数の決定がうまく行なわれていない。例として表4.7の関数を与えた場合の合成結果を図4.4に示す。

表4.7 4値2変数関数 $f_3(x,y)$

$y \backslash x$	0	1	2	3
0	3	0	2	1
1	1	2	0	3
2	0	3	1	2
3	2	1	3	0

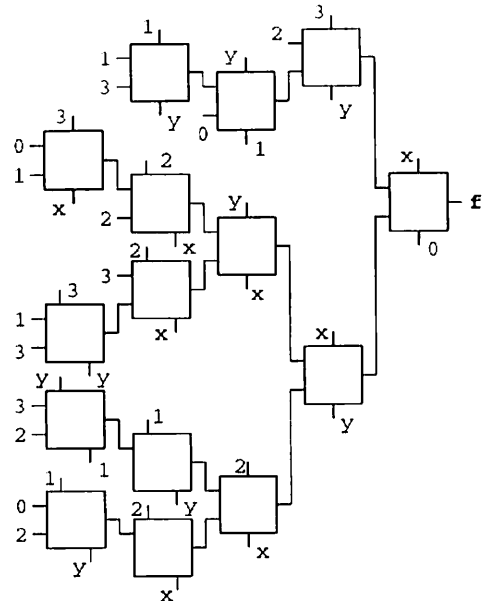


図4.5 関数 $f_3(x,y)$ のSVT回路網

表4.7の関数は、1段目のSVTゲートのしきい値制御側に3個のゲートを、2段目のゲートに1個のゲートを接続することにより、12個のSVTゲートを用いて合成できるが、シミュレーション結果で得られた最適解は15個のSVTゲートを使用している。これは、最適なしきい値関数の決定がうまく行なわれていないため、局所解に陥ったものである。

5. まとめ

本研究では、遺伝子長可変型GAを用いた4値2変数関数SVTゲート回路網の合成及びその合成に関する処理について述べた。そして、その回路網合成アルゴリズムをC言語でプログラミングし、一様乱数で生成した4値2変数関数を合成するシミュレーション結果を示した。

今回提案した新しい突然変異により、より多様なしきい値関数の生成が可能となり、ゲート数の減少が図られるようになった。

しかしながら、まだしきい値関数の決定が上手くいかず、最適解に達していない4値2変数関数も幾つか存在する。

そのため今後の改良点として、各パラメータの最適値の決定、新しい処理や適応度の設定などが考えられる。

また、出力側からみて1段目のSVTゲートのしきい値制御・制御端子のみに対して行なう突然変異2を、2段目のSVTゲートにも適用させ、そのシミュレーションを行なう必要があると思われる。

参考文献

- [1] 安居院 猛・長尾 智晴：「ジェネティックアルゴリズム」：昭晃堂
- [2] 伊庭 斉志：「遺伝的プログラミング」：東京電機大学出版局
- [3] 翁長 彰子：「平成7年度卒業論文 遺伝的アルゴリズムを用いた4値SVTゲート回路網の合成」
- [4] 新垣 学：「平成6年度修士論文 単一可変しきい値ゲートと4値論理関数への応用」
- [5] 瑞慶覧 長定・山城 哲哉・安富祖 忠信：「4値可変しきい値論理ゲートとその数学的性質」琉球大学工学部紀要, Vol.44, pp.129-135, 1992
- [6] 瑞慶覧 長定・島袋 勝彦・小波津 清正：「遺伝子長可変型GAによる4値VTゲート回路網の合成」, 日本ファジィ学会九州支部7月研究例会 No. 96-7-1 (1996-07)