免疫的最適化手法 —VLSI フロアプラン設計問題への応用—

学生員 田 澤 功 (千葉大)

正 員 小 坏 成 一 (千葉大)

正 員 平 田 廣 則 (千葉大)

An Evolutionary Optimization based on the Immune System and its Application to the VLSI Floorplan Design Problem

Isao Tazawa, Student Member, Seiichi Koakutsu, Member, and Hironori Hirata, Member
(Chiba University)

Genetic Algorithms (GAs) are search procedures for combinatorial optimization problems. Because GAs are based on multi-point search and use crossover operator, GAs have an excellent global search ability. However GAs are not effective for searching the solution space locally due to crossover-based-search, and the diversity of the population sometimes decreases rapidly. In order to overcome these drawbacks, we propose a new algorithm called Immunity-based GA (IGA) combining features of the Immune System with GAs. IGA is expected to improve the local search ability of GAs and maintain the diversity of the population. We apply IGA to the VLSI floorplan design problem. Experimental results show that IGA performs better than GAs.

キーワード：免疫システム，遺伝的アルゴリズム，VLSI フロアプラン設計問題

1. はじめに

近年，遺伝的アルゴリズム（以下，GA：Genetic Algorithm）^{(1)~(3)}の組合せ最適化問題への応用が盛んに行われている。GAでは，常に複数の解を保持し，主として二つの解を組み合わせる交差オペレータにより新しい解を生成するため，状態空間を広く探索することができる。しかし，GAを用いた組合せ最適化手法では，初期収束と呼ばれる問題と局所的（近傍）探索能力の欠如が指摘されている。初期収束とは，探索の初期段階で他の解に比べ優れた解が生成された場合に，その解が集団中に多く残り解集団の多様性が急速に失われる現象である。その結果，局所最適解に収束し，以降の探索が停止する。また，GAで用いられる交差は状態空間上で大きなジャンプを生成する傾向にあるため，ある解の近傍での探索が十分に行われない。

以上のようなGAの問題点を改善するために，分散型GA⁽⁴⁾⁽⁵⁾やハイブリッドGA⁽²⁾が提案されている。分散型GAは，解集団を複数の部分集団に分割し，個々の部分集団においてGAを実行する並列処理型のGAである。分散型GAは，初期収束を回避し解集団の多様性を維持することができるものの，局所的探索能力はGAと同程度であると考えられる。ハイブリッドGAは，GAにヒュー

リスティックな局所探索法を組み合わせることによりGAの局所的探索能力を向上させる手法である。しかし，初期収束については特に考慮されていない。これらの改良型のGAとは異なり，免疫システム⁽³⁾⁽⁶⁾⁽⁷⁾に注目した新しい最適化手法がいくつか提案されている。Bersini⁽⁸⁾⁽⁹⁾らは，免疫補充テストと呼ばれる解の選択方法をGAに導入することにより，局所的探索能力の向上をはかるGIRMを提案している。また，GIRMの応用例⁽¹⁰⁾も報告されている。しかし，GIRMでは初期収束に対する処置は行われていない。他方，森⁽¹¹⁾らは，免疫システムのみに注目して免疫アルゴリズム（IA）を提案している。IAは，GAのsharing法⁽¹²⁾に似た方法で初期収束を避け解の多様性を維持している。また，IAでは解の評価関数値を用いた親和度関数を定義して解の選択に使用することにより，局所的探索を促進している。しかし，本来，親和度は抗原と抗体の結合の強さを表す指標であるため，解の評価関数値ではなく解の構造に基づいて定義するのが望ましい。

本論文では，免疫システムのクローン選択機構とイデオタイプネットワークをGAに組み合わせた免疫的最適化手法（以下，IGA：Immunity-based GA）を提案する。クローン選択機構は，特定の抗体量を増加させる機構であり，イデオタイプネットワークは，抗体量の調節機構である。IGAでは，GAと同様に解集団を形成して，交差オ

ペレータや突然変異オペレータを用いて新しい解を生成する。そして、局所的探索性能の向上を図るため、解集団に残す解を選択する際に、免疫システムにおけるクローン選択機構を模倣した選択方法を使用する。その際、IGAでは、解の構造の類似性を表す類似度関数を定義して解の評価に使用する。また、解の多様性を維持するために、集団をいくつかの部分集団に分割し、部分集団の大きさの調整に免疫ネットワーク的な機構を使用する。

本論文では、提案手法の有効性を検証するために、すでにGAが適用されているVLSIフロアプラン設計問題にIGAを応用し、GAにより得られた結果と比較する。類似度関数をモジュールの幾何学的な配置に基づき定義し、IGAをフロアプラン設計問題に応用した結果、GAで得られる解より優れた解がより高い確率で得られることが示された。

2. 免疫的最適化手法

図1に免疫システムの概略図を示す。免疫は、生体内に侵入した異物に刺激を受けたリンパ系細胞が相互に影響して起こる反応のことである。このような反応を引き起こす異物のことを抗原 (Antigen) と呼ぶ。抗原が生体内に侵入すると、その抗原を排除するために、免疫システムは抗体 (Antibody) を産生する。抗体はリンパ系細胞の一つであるB細胞 (B cell) により産生され、抗原と結合して抗原を無毒化したり、他の免疫担当細胞に抗原を認識させる役割を果たす。ある一つの種類の抗体が対処できる抗原の種類は限られているので、様々な種類の抗原を排除するために多様な抗体が産生される。抗原と抗体の適合の度合い (結合の強さ) を表す尺度のことを親和性と言い、抗原と高い親和性を有する抗体のみが抗原に対処できる。

本論文では、免疫システムの二つの主要な機構に注目する。一つは抗体の産生機構であり、もう一つは抗体の濃度の調整機構である。抗体の産生機構であるクローン選択機構は、抗原と高い親和性を有する特定の抗体を産生するB

細胞を選択的に増殖する機構である。例えば、図1では抗原と親和性の高い抗体 A_1 を産生する B_1 が増殖する。その結果、 A_1 の量が増加し抗原に対処する。

選択的に増殖したB細胞の産生する抗体により抗原の排除が行われるが、一方では一種類の抗体が過剰に産生されるのを防ぐために、抗体産生を停止するような抑制が働く。その一つにイディオタイプネットワークによる抑制がある。これはB細胞のクローンにより構成される閉鎖的なネットワークであり、増殖した抗体自体が抗原となり、それに結合できる抗体が産生されて増殖した抗体の産生を抑制する。例えば、図1では A_2 が増加して A_1 の増加を抑え、更に A_2 の増加を A_3 が抑制する。

以下、上記の免疫システムに基づいた提案手法について説明する。

〈2・1〉 免疫的最適化手法のアルゴリズム

本論文では、免疫システムのクローン選択機構およびイディオタイプネットワークをもとにGAを改良した、免疫的最適化手法 (以下、IGA: Immunity based GA) を提案する。

まず、GAの局所的探索性能を改善するために、GAとクローン選択機構を組み合わせる。そのため、二つの解の類似している度合いを表す類似度関数を定義する。類似度関数は二つの解に対して定義され、抗原と抗体、あるいは抗体同士の親和性に対応する。そして、集団内で適応度が大きな解 (一つ、または複数個) を抗原、その他の解を抗体とみなし、類似度に基づいて次世代を形成する解を選択する。類似度に基づいて解を選択することにより、適応度の大きな解に類似した解の個数が増加するため、解空間上で適応度の大きな解の周辺部の探索が促進されることになる。

次に、初期収束の問題を解決するために、GAとイディオタイプネットワークを組み合わせる。まず、解集団を複数の部分集団に分割する。各部分集団は、実際の免疫システムにおいて異なる抗原と特異的に反応し、互いに刺激、抑制し合うB細胞のクローン (もしくはその産生する抗体) に対応する。そして、イディオタイプネットワークにより、部分集団内の解の多様性と部分集団間の類似度を考慮して部分集団の大きさを調整する。具体的には、部分集団の大きさは、部分集団間で解を移動させることにより調整する。すなわち、部分集団内の解の個数は変動するが、すべての部分集団内の解の個数の総和は一定であるとする。

図2に、IGAのアルゴリズムを示す。IGAでは、まず初期集団 P をアトランダムに生成し (1:), これらを複数の部分集団 $P_i (i=1, 2, \dots, N_s)$ に分割する (2:)。初期部分集団を形成した後、交差オペレータおよび突然変異オペレータを使用して、部分集団ごとに新しい解を生成する (5: , 6:)。部分集団ごとに決められた個数の解を生成した後、部分集団ごとに次世代を形成する解を類似度に基づいて選択する (7: , 8:)。そして、世代数がある世代数 N_m に達したときにすべての部分集団の大きさを調整する

図1 免疫システムの概略図

Fig. 1. An illustration of the Immune System.

```

1: initialize a population  $P$  of solutions;
2: divide  $P$  into a set of subpopulations  $\{P_1, \dots, P_{N_s}\}$ ;
3: while generation_number <  $N_g$  do
4:   parfor subpopulation  $P_i$  do
5:     generate  $r_c \cdot N_i$  solutions using crossover;
6:     perturb  $r_m \cdot N_i$  solutions using mutation;
7:     select  $N_i$  solutions
8:       according to their affinities;
9:   endfor
10:  if generation number =  $N'_m$  then
11:    adjust the size of each subpopulation;
12:     $N'_m = N_m$ ;
13:    generation_number ++;
14: endwhile

```

図 2 IGA のアルゴリズム
Fig. 2. Algorithm of IGA.

(11:)。4: から 13: までは一世代とし、決められた世代数 N_g だけ上記の処理を繰り返す。

〈2・2〉 類似度関数 類似度関数⁽⁹⁾は、二つの解の距離を用いて以下のように定義される。

$$a(x_i, x_k) = 1 - \frac{d(x_i, x_k)}{d_{\max}} \quad (1)$$

ここで、 $a(x_i, x_k)$ は解 x_i, x_k 間の類似度、 $d(x_i, x_k)$ は解 x_i, x_k 間の距離、 $d_{\max}$ は $d(x_i, x_k)$ の最大値である。(1) 式から明らかなように類似度関数 $a(x_i, x_k)$ の値は 0 以上 1 以下であり、二つの解の類似性が高いほど 1 に近い値となる。

〈2・3〉 類似度に基づいた解の選択 IGA では、すべての解に対して次式(2)の値を計算する。そして、その値がより大きな解を高い確率で集団に残して新しい集団を形成する。

$$s(x_i) = \frac{\sum_{k=1}^{r_a \cdot N_j} a(x_i, x_k) f(x_k)}{\sum_{k=1}^{r_a \cdot N_j} a(x_i, x_k)} \quad (2)$$

ここで、 N_j は解 x_i の属する部分集団 P_j の大きさ、 r_a は集団内で抗原とみなされる解の割合、 $f(x_k)$ は解 x_k の適応度である。 $r_a \cdot N_j$ 個の解は、集団内で適応度が大きな解から順に選択する。

$s(x_i)$ は x_i が適応度の大きな解(抗原)に類似しているほど大きくなる。そのため、(2)式に基づいて解の選択を行うことにより解の探索は解集団内の適応度の高い解の付近に限定され、局所的な探索を活発に行うことができる。部分集団内に類似した解が増加するため、この操作は実際の免疫システムのクローン選択機構に対応しているといえる。

〈2・4〉 部分集団の大きさの調整 まず、部分集団を構成する解の個数およびそれらの評価関数値の分散から、世代 t における部分集団 P_i の有効な大きさ $\hat{N}_i(t)$ を以下のように定義する。

$$\hat{N}_i(t) = \frac{\sigma_i^2}{\sigma_{\max}^2} N_i(t) \quad (3)$$

ここで、 σ_i^2 は部分集団 P_i 内の解の評価関数値の分散、

図 3 部分集団間の相互作用
Fig. 3. An illustration of the interaction between subpopulations.

$\sigma_{\max}^2$ は σ_i^2 の最大値、 $N_i(t)$ は部分集団 P_i を構成する解の個数である。

$\hat{N}_i(t)$ は部分集団内の解の多様性を表している。すなわち、 $\hat{N}_i(t)$ が大きな場合には、部分集団内に多くの異なる解が存在し、 $\hat{N}_i(t)$ が小さな場合には、部分集団内に類似した解が多数存在することになる。

次に、(3)式で定義した集団の有効な大きさと部分集団の類似度を用いて、部分集団 P_i の大きさの更新式を以下のように定義する。

$$N_i(t+1) = N_i(t) + N_a \frac{\Delta \hat{N}_i}{\frac{1}{2} \sum_{j=1}^n |\Delta \hat{N}_j|} \quad (4)$$

$$\Delta \hat{N}_i = \sum_{j=1}^{N_s} \{1 - A(P_i, P_j)\} \{\hat{N}_j(t) - \hat{N}_i(t)\} \quad (5)$$

$$A(P_i, P_j) = a(X_i, X_j) \quad (6)$$

ここで、 N_a は部分集団の大きさの変化分の最大値、 N_s は部分集団の個数である。 $A(P_i, P_j)$ は部分集団 P_i, P_j 間の類似度であり、部分集団 P_i, P_j 内で適応度が最大である解 X_i, X_j の類似度として定義する。

図 3 に部分集団間の相互作用の例 ($N_s=4, i=1$) を示す。部分集団 P_i の大きさの変化分 $\Delta \hat{N}_i$ は、 P_i の有効な大きさ $\hat{N}_i(t)$ が他の部分集団(図 3 では $P_j (j=2, 3, 4)$) の $\hat{N}_j(t)$ に比べて小さいとき正になり、逆に $\hat{N}_i(t)$ が $\hat{N}_j(t)$ に比べて大きいとき負になる。また、 $\Delta \hat{N}_i$ は P_i との類似度が小さい部分集団との大きさの差 ($\hat{N}_j(t) - \hat{N}_i(t)$) に大きく依存する。すなわち、 $\hat{N}_i(t)$ が小さく、 P_i 内の解が他の部分集団内の解と類似していないとき、 $\Delta \hat{N}_i$ は大きな正数となり、他の部分集団から P_i への解の移動が起こる。その結果、 P_i 内の解の多様性が増すので多様な解の生成が促進される。逆に、 P_i 内の解が他の部分集団内の解と類似しているときには、 $\Delta \hat{N}_i$ は小さな正数となり P_i の大きさはほとんど変化しない。これは、 $\hat{N}_i(t)$ が大きく $\Delta \hat{N}_i$ が負となる場合と同様に、類似した解の過剰な増加の抑制を意味する。以上のように、(4)式を用いて解集団全体の多様性を維持することで初期収束を回避できる。部分集団間の類似度と部分集団の大きさに基づいて互いの大きさを調整するので、この操作は実際の免疫システムの抗体産生

3. VLSI フロアプラン設計問題への適用

〈3・1〉 VLSI フロアプラン設計問題 フロアプラン設計問題とは、配置するべきすべての回路素子（モジュール）がチップ上にのるように、チップ領域を分割する問題である。すでに GA の適用されているフロアプラン設計問題は以下のように定義されている⁽¹³⁾。本論文で扱うフロアプラン設計問題も同様に定義する。

- スライス構造をもつチップの例を図4(a)に示す。スライス構造とは、水平および垂直線分のみで得られる分割構造である。スライス構造のフロアプランは、(b)図に示すような木(分割木)で表すことができる。「+」は水平な境界、「*」は垂直な境界を表し、数字はチップ上の各領域に収容されるモジュールを表す。

得られたフロアプラン(解)の評価には、チップの総面積とモジュール間の仮想配線長から定義される評価関数を使用する。仮想配線長とは、モジュールの中心間のマンハッタン距離とそのモジュール間の配線数の積である。評価関数を以下に示す。

ここで、 n はモジュール数、 W_i , H_j はチップ上の分割された各領域の幅および高さ、 c_{jk} , d_{jk} はモジュール j , k 間の配線数および中心間距離、 λ は重み定数である。フロアプラン設計では、(7)式の評価関数値が最小となるような配置を求める。

交差では、適応度に基づいて解集団から二つの解(親) P_{a1} , P_{a2} を選択し、図5に示す三つの操作のうちの一つを行うことで新しい解(子) O を生成する。C1では、まず

図 5 交差オペレータの例
Fig. 5. An illustration of the crossover operators.

図 6 突然変異オペレータの例
Fig. 6. An illustration of the mutation operators.

P_{a1} の数字を $O1$ の対応する位置にコピーして、次に P_{a2} の符号を左から順に $O1$ が完成するようにコピーする。
C1により、 $O1$ は P_{a1} の数字を逆ポーランド表記上ではそのままの形で受け継ぐ。C2では、まず P_{a1} の符号を $O2$ の対応する位置にコピーして、次に P_{a2} の数字を左から順に $O2$ が完成するようにコピーする。C2により、 $O2$ は P_{a1} の分割構造のみを受け継ぐ。C3では、まず P_{a1} の符号を $O3$ の対応する位置にコピーする。次に P_{a1} の部分木の一つ選択して、その数字を $O3$ の対応する位置にコピーする。そして、 P_{a2} の数字を左から順に(2)のものと重複しないように $O3$ にコピーする。C3により、 $O3$ は P_{a1} の配置の一部分をそのまま受け継ぐ。

突然変異では、一つの解を任意に選択し、図6に示す三つの操作のうちの一つを行うことで新しい解を生成する。
M1では任意の二つの数字を交換する。そのため、チップ上で二つのモジュールの位置が入れ替わる。M2では、任意の一つの符号(＊、＋)を反転させる。そのため、複数のモジュールより形成されているブロックの分割形状は変化しないが、二つのブロック同士の位置関係もしくは二つのモジュールの位置関係が変化する。M3では、隣接する符号と数字を交換する。そのため、チップの分割構造が大きく変化する。

解の評価には、(7)式の評価関数に基づいて定義される適応度関数を使用する。適応度関数は、評価関数値が小さいほど値が大きくなる関数である。本論文で使用する適応度関数⁽¹³⁾を以下に示す。

$$f(x_i)=\frac{\{\mu_s-c(x_i)\}+\sigma_s}{2\sigma_s} \dots\dots\dots (8)$$

ここで、 μ_s , σ_s はすべての解の評価関数値の平均値および標準偏差である。 $f(x_i)$ が負となる場合には、 $f(x_i)$ の値は小さな正の数とする。

〈3・3〉 類似度関数の定義 各モジュールの中心座標(チップ上の位置)、幅および高さを用いて、二つの解の類似性を表す類似度関数を次のように定義する。

$$d(x_i, x_k)=d_1(x_i, x_k)+d_2(x_i, x_k) \dots\dots\dots (9)$$

$$d_1(x_i, x_k)=\sum_{j=1}^n\sqrt{(u_{ij}-u_{kj})^2+(v_{ij}-v_{kj})^2} \dots\dots\dots (10)$$

$$d_2(x_i, x_k)=\sum_{j=1}^n\sqrt{(w_{ij}-w_{kj})^2+(h_{ij}-h_{kj})^2} \dots\dots\dots (11)$$

ここで、 w_{ij} , h_{ij} は解 x_i の j 番目のモジュールの幅および高さ、 (u_{ij}, v_{ij}) は解 x_i の j 番目のモジュールの中心座標である。

4. 計算機実験結果

表1に実験で使用するフロアプラン設計問題を示す。
 $Ami33$ は、モジュール数33、ネット数123、 $Ami49$ は、モジュール数49、ネット数408のブロック配置問題であり、MCNCベンチマークデータ⁽¹⁴⁾の一つである。実験で使用する際には、各モジュールの幅および高さは可変であるとし、また、多端子ネットはすべてモジュール間の二端

表 1 フロアプラン設計問題
Table 1. Floorplan instances.

	<i>Ami33</i>	<i>Ami49</i>
# of modules	33	49
# of nets	123	408

表 2 各種パラメータの値
Table 2. Values of parameters.

Parameter	Symbol	Value		
		IGA	GA	GIRM
Size of population	N	200	200	200
Crossover rate	r_c	0.8	0.8	0.8
Mutation rate	r_m	0.3	0.3	0.3
# of subpopulation	N_s	4	-	-
Rate of antigens in each subpopulation	r_a	0.05	-	-
Maximum variation of the subpopulation size	N_a	12	-	-
Generation gap for remove of solutions	N_m	100	-	-

図 7 各世代における $Ami33$ の解の評価関数値
Fig. 7. An example of the convergence of the cost.

子ネットに変換する。

比較対象としてはGAおよびGIRM⁽¹⁰⁾を使用する。
GIRMで使用する類似度としては、IGAと同様に(1)式、(9)式で定義される類似度を使用する。表2に実験で使用する各種パラメータの値を示す。 $Ami33$ を使用する場合は5,000世代まで実験を行い、また、 $Ami49$ を使用する場合は20,000世代まで実験を行う。図7に $Ami33$ の解の収束の様子を例を示す。図中で「-OPT」、「-AVE」は、それぞれ解集団内の解の評価関数値の最小値と平均値の変化を表している。図7の「-OPT」より、5,000世代までに解は十分に収束することがわかる。同様に、 $Ami49$ では20,000世代までに解が十分に収束する。なお、各手法の最終的な解は、最終世代における解集団内の最良解とする。

計算機実験には、SPARC Station 20クラスのワークステーションを使用する。初期解を変えて100回実験を行った結果を表3に示す。表中の値は、各手法で得られた最良解の評価関数値(Best)、平均値(Average)および平均

表 3 各手法による実験結果
Table 3. Experimental results.

		Method		
		IGA	GIRM	GA
Ami33	Best	4.83×10^6	4.83×10^6	4.85×10^6
	Average	4.93×10^6	5.00×10^6	5.02×10^6
	Time (s)	1,006	2,730	773
Ami49	Best	3.80×10^7	3.80×10^7	3.80×10^7
	Average	3.85×10^7	3.86×10^7	3.89×10^7
	Time (s)	13,967	25,122	12,334

図 8 IGAおよびGAにより得られたAmi33 の解の頻度分布
Fig. 8. A frequency distribution of 100 solutions of Ami33 obtained by IGA and GA.

図 9 IGAおよびGAにより得られたAmi49 の解の頻度分布
Fig. 9. A frequency distribution of 100 solutions of Ami49 obtained by IGA and GA.

計算時間 (Time) である。また、図 8 から図 11 に 100 回の試行で得られた解の評価関数値の頻度分布を示す。図 12、図 13 は、IGA により得られた Ami33 および Ami49 のフロアプランの例である。

表 3 より IGA と GA を比較すると、Ami33 については、IGA でより評価関数値の小さな優れた解が得られており、また、平均的にみても IGA の方が GA より優れた解を生成していることがわかる。Ami49 については、得られた最良値は等しいが、IGA では平均的により優れた解を生成していることがわかる。また、IGA の計算時間は GA より若干増加している。次に、IGA と GIRM を比較すると、得られた最良値は等しいが、IGA では平均的

図 10 IGA および GIRM により得られた Ami33 の解の頻度分布
Fig. 10. A frequency distribution of 100 solutions of Ami33 obtained by IGA and GIRM.

図 11 IGA および GIRM により得られた Ami49 の解の頻度分布
Fig. 11. A frequency distribution of 100 solutions of Ami49 obtained by IGA and GIRM.

により優れた解を生成していることがわかる。また、IGA の計算時間は GIRM の 50%程度に短縮されている。

図 8、図 9 より GA に比べて IGA で良い解が多く得られており、得られた解の評価関数値の分散も IGA の方が小さいといえる。Ami33、Ami49 のどちらの問題についても、GA では IGA で得られた解の評価関数値の平均値より小さな評価関数値をもつ解は、100 回の試行で 10%程度しか得られていない。また、図 10、図 11 より、Ami33 については GIRM に比べて IGA で良い解が多く得られており、得られた解の評価関数値の分散も IGA の方が小さいことがわかる。GIRM では IGA で得られた解の評価関数値の平均値より小さな評価関数値をもつ解は、100 回の試行で 20%程度しか得られていない。Ami49 については、IGA の方が分散は小さいが、IGA の平均値より小さな評価関数値をもつ解は、IGA と GIRM で同程度得られている。

以上の結果から、IGA による探索は GA による探索に比べて局所解に陥りにくく、また IGA が GA に比べ局的探索をより活発に行っていると考えられる。また、IGA と GIRM の局所探索能力は同程度であるが、IGA は GIRM に比べて局所解に陥りにくく、その結果、IGA の方がより有効に探索を行えると考えられる。

図 12 IGA により得られた Ami33 のフロアプランの例

Fig. 12. An example of the floorplan of Ami33 generated by IGA.

図 13 IGA により得られた Ami49 のフロアプランの例

Fig. 13. An example of the floorplan of Ami49 generated by IGA.

図 7 より、GA および GIRM は IGA に比べより少ない世代数で急速に収束していることがわかる。また、解集団内の解の評価関数値の平均値を比べると、IGA の方が大きく保たれていることがわかる。このことから、GA や GIRM に比べて IGA は解集団の多様性を保ちながら解の探索を行っているといえる。そのため、更に世代数を増加させると GA や GIRM に比べ更に優れた解が得られる可能性があると考えられる。

5. おわりに

本論文では、免疫システムにおけるクローン選択機構と抗体産生の自己調節機構（イディオタイプネットワーク）に基づく免疫的最適化手法（IGA）を提案した。IGA では、解集団を複数の部分集団に分割して部分集団ごとに解の生成、選択を行う。解の生成には GA と同様に交差オペレータや突然変異オペレータを使用し、適応度の大きな解に類似した解を選択的に残す。そして、部分集団内の解

の評価関数値の分散や部分集団間の類似度に基づいて、部分集団の大きさの決定および解の移動を行う。そのため、解集団の多様性を維持しながら局所的探索をより活発に行うことができる。

また、本論文では IGA の有効性を検証することを目的に、すでに GA が適用されている VLSI フロアプラン設計問題に IGA を適用した。IGA を実際に問題に適用する場合、解の類似性を表す類似度関数を定義する必要がある。今回はモジュールの幾何学的な配置に基づき定義した。この類似度関数を使用して IGA をフロアプラン設計問題に適用した結果、GA で得られる解より優れた解がより高い確率で得られることが示された。

現段階では、IGA における解の生成方法として GA と同様に交差を使用している。しかし、実際の免疫システムでは抗体の多様性を生み出すために、GA における交差オペレータとは異なる遺伝子操作が行われている。そのような遺伝子操作を IGA に取り入れ、より多様な解を生成、保持しながら、より優れた解を探索する手法の開発が今後の課題としてあげられる。

文 献

- (1) D. E. Goldberg: "Genetic Algorithms in search, optimization and machine learning", Addison-Wesley Publishing Company, Inc. (1989)
- (2) 北野宏明: 遺伝的アルゴリズム, (平 5) 産業図書
- (3) 北野宏明: 遺伝的アルゴリズム 2, (平 7) 産業図書
- (4) J. Nang, & K. Matsuo: "A Survey on the Parallel Genetic Algorithms", 「計測と制御」, 33, No. 6, 500~509 (平 6)
- (5) 玉置 久, 喜多 一, 岩本貴司: 「遺伝的アルゴリズム-V-GA の拡張」, システム/制御/情報, 40, No. 2, 69~76 (平 8)
- (6) J. D. Farmer, N. H. Packard, & A. S. Perelson: "The Immune System, Adaptation, and Machine Learning", Physica 22D, 187~204 (1986)
- (7) 和田健之介, 田中伸一, 和田佳子: 「進化システム論 遺伝的アルゴリズムの基礎(5)」, in Computer Today, 51, No. 9, 82~87 (平 4)
- (8) H. Bersini & F. J. Varela: "Hints for Adaptive Problem Solving Gleaned from Immune Networks", Proc. 1st Workshop of PPSN, 343~354 (1990)
- (9) H. Bersini & F. J. Varela: "The Immune Recruitment Mechanism: A Selective Evolutionary Strategy", Proc. 4th Int. Conf. on Genetic Algorithms, 520~526 (1991)
- (10) 田澤 功, 小坪成一, 平田廣則: 「免疫機構を取り入れた遺伝的アルゴリズムの VLSI フロアプラン設計への応用」, 計測自動制御学会論文集, 31, No. 5, 615~621 (平 7)
- (11) 森 一之, 築山 誠, 福田豊生: 「多様性を持つ免疫的アルゴリズムの提案と負荷割当問題への応用」, 電学論 C, 113, 872~878 (平 5-10)
- (12) D. E. Goldberg & J. Richardson: "Genetic Algorithms with Sharing for Multimodal Function Optimization", Proc. 2nd Int. Conf. on Genetic Algorithms, 41~49 (1987)
- (13) J. P. Cohoon, S. U. Hedge, W. N. Martin, & D. S. Richards: "Distributed Genetic Algorithms for the Floorplan Design Problem", IEEE Trans. CAD, 10, No. 4, 483~492 (1991)
- (14) K. Kozminski: "Benchmarks for Layout Synthesis-Evolution and Current Status", int. Proc. 28th ACM-IEEE Design Automation Conf., San Francisco, CA, 265~270 (1991)

田 澤 功 (学生員) 1971 年 3 月 14 日生。95 年千葉大学大学院工学研究科電気電子工学専攻修士課程修了。同年同大学院自然科学研究科生産科学専攻博士課程に入学，現在に至る。確率的最適化手法，VLSI レイアウト設計の研究に従事。電子情報通信学会会員。

小 坏 成 一 (正員) 1964 年 7 月 7 日生。92 年千葉大学大学院自然科学研究科生産科学専攻博士課程修了。同年千葉大学工学部助手となり，現在に至る。1994～95 年，カリフォルニア大学サンタクルズ校客員研究員。VLSI レイアウト設計，確率的最適化手法の研究に従事。博士 (工学)。IEEE，INNS，電子情報通信学会，計測自動制御学会会員。

平 田 廣 則 (正員) 1948 年 6 月 2 日生。76 年東京工業大学大学院電気工学専攻博士課程修了。同年千葉大学工学部助手，81 年同助教授，94 年同教授，97 年千葉大学大学院自然科学研究科教授となり現在に至る。大規模システムのモデリング，解析と設計，特に生態システム，VLSI レイアウト，並びに分散型コンピュータの基礎的理論に興味をもつ。工学博士。IEEE，ISEM，INNS，電子情報通信学会，情報処理学会，計測自動制御学会，システム制御情報学会会員。