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

田 澤 功*・小 坪 成 一**・平 田 廣 則**

A VLSI Floorplan Design Based on Genetic Immune Recruitment Mechanism

Isao TAZAWA*, Seiichi KOAKUTSU** and Hironori HIRATA**

Genetic algorithms (GAs) are search procedures for combinatorial optimization problems. Unlike most of other optimization techniques, GAs search the solution space using a population of solutions. Although GAs have an excellent global search ability, it is not effective for searching the solution space locally due to crossover-based search. Genetic Immune Recruitment Mechanism (GIRM) makes up for the weak point of GAs. In GIRM, new generated solutions take an immune recruitment test. Only those solutions which are similar to the best solution in the population pass the test and survive to the next generation. As a result, GIRM promotes the search around the best solution. In this paper we apply GIRM to the floorplan design problem of VLSI layout and compare the results with the ones of GAs. Especially, we propose new affinity functions in order to evaluate the similarity between floorplans. Computational experiments show that GIRM gives equivalent or better solutions than GAs.

Key Words: genetic algorithms, genetic immune recruitment mechanism, affinity function, immune recruitment test, floorplan design

1. はじめに

フロアプラン設計問題は、ある制約条件下でモジュールの最適配置状態を求める問題であり、一種の組合せ最適化問題とみなすことができる^{1),2)}。このような組合せ最適化問題を解くために、遺伝的アルゴリズム (Genetic Algorithms, 以下, GA)^{3)~5)}の適用が提案されている。GAは、生物集団の進化の過程をモデルにしており、多点探索を行う。そのため、探索範囲が広く、また局所解に陥りにくいとされている。しかし、解の局所的探索性能は低いとされている⁶⁾。

近年、免疫系の理論的解析が進み、GAと免疫システムの関連性の指摘や、免疫系の理論の組合せ最適化問題へ

の応用がなされている^{7)~9)}。その一つとして、局所的探索を補うためGAに免疫の仕組みを取り入れた手法 (Genetic Immune Recruitment Mechanism, 以下, GIRM)^{6),10)~12)}がある。GIRMでは、淘汰のかわりに免疫産生テストを導入している。免疫産生テストは、適応度の高い解との類似性を問うものであり、テストを満足する解は解集団で最も適応度の低い解と入れ換えられる。その結果、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、局所的な探索が活発に行われることになる。

本論文では、GIRMの有効性を検証することを目的に、すでにGAが適用されているフロアプラン設計問題にGIRMを適用し、GAにより得られた結果との比較を行う。また、GIRMを適用する際に必要な類似度関数を、フロアプラン設計問題について2種類定義する。一つは、モジュールの幾何学的な配置、すなわち表現型に基づき定義する。もう一つは、コーディングされた遺伝子列、すなわち遺伝子型に対して定義する。最適値が既知である問題に適用した結果、GIRMでは、GAで得られた解と同等かそれ以上の解がより高い確率で得られることが示された。

* 千葉大学大学院自然科学研究科 千葉市稲毛区弥生町1-33

** 千葉大学工学部 千葉市稲毛区弥生町1-33

* Graduate School of Science and Technology, Chiba University, Chiba

** Faculty of Engineering, Chiba University, Chiba
(Received September 13, 1994)
(Revised December 1, 1994)

2. 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ことである。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^{1),2)}。本論文で扱うフロアプラン設計問題も同様に定義する。

- 配置すべきモジュールは長方形であり、その面積は一定である。
- モジュールの幅 w_i および高さ h_i は可変であり、その比(アスペクト比)の上限と下限は決められている。
- チップはスライス構造をもつ。
- チップはモジュールを完全に含む、幅 x_i 、高さ y_i の長方形の領域にすまなく分割される($w_i \leq x_i$, $h_i \leq y_i$)。

スライス構造をもつチップの例を Fig. 1(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、Fig. 1(b) に示すような木(分割木)で表わすことができる。「+」は水平な境界、「*」は垂直な境界を表わし、数字はチップ上の各領域に収容されるモジュールを表わす。

今、二つのスライス構造 A, B を考え、 A の上に B が位置することを「 $AB+$ 」、 A の右に B が位置することを「 $AB*$ 」で表現することにする。Fig. 1(b) の分割木は、左から右そして下から上へと処理することにより

$$12*34+5*+ \quad (1)$$

と表わすことができる。分割木は、(1) 式のような逆ポーランド表記と呼ばれる文字列と 1 対 1 に対応する。

フロアプラン設計で得られた配置結果(解)の評価には、チップの面積とモジュール間の配線長から定める評価関数を用いる。配線長は、モジュールの中心間のマンハッタン距離に重みをかけたものである。評価関数を以下に示す。

$$c(x) = \sum_{i=1}^n x_i y_i + \lambda \sum_{i,j=1}^n c_{ij} d_{ij} \quad (2)$$

ここで、 n はモジュール数、 c_{ij} はモジュール i, j 間の配線数、 d_{ij} は中心間距離、 x_i, y_i はチップ上の分割され

Fig. 1 A slicing structure and a slicing tree

た各領域の幅および高さ、 λ は定数である。フロアプラン設計では、(2) 式の評価関数が最小となるような配置を求める。

3. 免疫機構を取り入れた遺伝的アルゴリズム

生体内に抗原が侵入すると、その抗原に対し最も適合性の高い抗体を産生するリンパ球のクローンが選択的に増殖する^{6),12),13)}。免疫学的手法(Immune Recruitment Mechanism, 以下、IRM)は、解をリンパ球もしくは抗体と考え上述の仕組みを取り入れたものである。IRM では、GA と同様に解集団を形成する。そして、適応度の大きい解に類似した解を生成、選択し、新しい解集団を形成する^{6),10)}。この IRM を GA の解の探索段階に取り入れることにより収束性の向上をはかった手法(以下 GIRM)が、提案されている^{6),10)}。GIRM では、解の生成(探索)を遺伝的オペレータを用いて行い、解の選択には免疫的選択手段を用いる。すなわち、GIRM における IRM の役割は、局所的探索を行うことではなく、適応度の高い解に類似した解を選択することにより、ある程度その周囲に探索を限定することである。類似した成分をもつ遺伝子列が集団内で増加するため、GA の交叉処理でも近傍探索が可能になることが期待される。

以下に GIRM のアルゴリズム^{6),12)}を示す。

- 初期解をランダムに発生させ、解の適応度を計算する。
- 確率的に以下の(a)または(b)の処理を行い、新しい解(子孫)を一つ生成する。
 - 集団から適応度の大きな解(親)を確率的に二つ選択し交叉を行う。
 - 集団からランダムに解を一つ選択し、突然変異を行う。
- 新しい解に対し免疫産生テストを課し、テストを満足する解を解集団に加える。代わりに集団内で最も適応度の低い解を除去する。
- 上記の処理(ii), (iii)を繰り返す。

本論文では、上記アルゴリズムの(ii)の時点で複数の解を生成する。そして、免疫産生テストを満足する解と既存の解から適応度の高い解を確率的に選択する。これにより、適応度の大きな解の近傍での解の探索がより活発になり、また、適応度の小さな解もある程度集団に残りやすくなることが期待される。さらに、後述の適応度を使用する場合、複数の解を生成し解の選択を行ったほうが、計算時間がかなり短縮される。

4. フロアプラン設計への適用

フロアプラン設計に GA や GIRM を適用する際、解は

逆ポーランド表記を用いて遺伝子コーディングする²⁾。すなわち、解は、数字および2種類の符号により表わされる。逆ポーランド表記で表わされる解に対する、交叉処理および突然変異処理を以下に示す。

交叉では、適応度に基づいて解集団から二つの解(親) P_1, P_2 を選択し、以下に示す操作²⁾のうちの一つを行うことで新しい解(子孫) O を生成する。

- C1: i. P_1 の数字を O の対応する位置にコピーする。
 ii. P_2 の符号を左から順に O が完成するようにコピーする。
- C2: i. P_1 の符号を O の対応する位置にコピーする。
 ii. P_2 の数字を左から順に O が完成するようにコピーする。
- C3: i. P_1 の符号を O の対応する位置にコピーする。
 ii. P_1 の部分木を一つ選択しその数字を O の対応する位置にコピーする。
 iii. P_2 の数字を左から順に(2)のものと重複しないように O にコピーする。

C1 では、子孫は P_1 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、子孫は P_1 の分割構造のみを受け継ぐ。C3 では、子孫は P_1 の配置の一部分をそのまま受け継ぐ。Fig. 2 に交叉の例を示す。

突然変異では、一つの解を任意に選択し、つぎの三つの操作^{1), 2)}のうちの一つを行うことで新しい解を生成する。Fig. 3 に突然変異の例を示す。

M1: 任意の二つの数字を交換する。

M2: 符号(*, +)を反転させる。

Fig. 2 An example of the crossover operator

M3: 隣接する符号と数字を交換する。

解の評価には、(2)式の評価関数に基づいて定義される適応度と呼ばれる関数を使用する。適応度は、評価関数値が小さいほどその値が大きくなる関数である。本論文では以下に示す適応度²⁾を用いる。

$$f(x_i) = \frac{(\mu_s - c(x_i)) + \sigma_s}{2\sigma_s} \quad (3)$$

ここで、 $f(x_i)$ は適応度、 μ_s はすべての解のコストの平均値、 σ_s は標準偏差である。 $f(x_i)$ が負となる場合には、 $f(x_i)$ の値は小さな正の数とする。

GIRM では解を評価するために、GA で使用する適応度のほかに類似度を使用する^{6), 10)}。類似度は二つの解に対し定義され、二つの解がどれくらい似ているかを表わす。類似度の値は0以上1以下であり、二つの解の類似性が大きいほど1に近づく。

本論文では、2種類の類似度を定義する。一つは、各モジュールの中心座標と幅および高さの差からつぎのように定義する。

$$m_l(i, k) = 1 - \frac{d_l(i, k)}{d_{ml}} \quad (4)$$

$$d_l(i, k) = \sum_j \sqrt{(a_{ij} - a_{kj})^2 + (b_{ij} - b_{kj})^2} \quad (5)$$

$$d_2(i, k) = \sum_j \sqrt{(w_{ij} - w_{kj})^2 + (h_{ij} - h_{kj})^2} \quad (6)$$

ここで、 $m_l(i, k)$ は解 x_i, x_k 間の類似度、 d_{ml} は $d_l(i, k)$ の最大値、 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、 (a_{ij}, b_{ij}) は解 x_i の j 番目のモジュールの中心座標である。 $d_l(i, k)$ は、 x_i, x_k 間の同一モジュールの中心座標に関する類似度であり、チップ上のモジュールの位置の類似性を表わす。 $d_2(i, k)$ は各モジュールの大きさの類似性を表わす。(5)式および(6)式を(4)式に代入し、モジュールの位置に関する類似度 m_1 およびモジュールの大きさに関する類似度 m_2 を求める。

上記の類似度は、遺伝子コードの表わすフロアプラン、すなわち表現型に対し定義している。これに対しもう一

Fig. 3 An example of the mutation operator

つの類似度は、逆ポーランド表記の遺伝子コード、すなわち遺伝子型に対し定義する。遺伝子コードにある操作を、二つのコードが等しくなるまで施し、その回数を二つの遺伝子コード x_i, x_k の距離 $d_3(i, k)$ とする。 $d_3(i, k)$ を(4)式に代入し、類似度 m_3 とする。遺伝子コードに施す操作を以下に示す¹⁾。

A1: 隣接した数字と符号の交換

A2: 連続した(長さ1以上の)符号の反転

本来ならば A1 と A2 を行う回数の最小値を距離とすべきである。しかし、最小回数を求めるのは困難であること、厳密ではなくてもある程度の類似性は求められることなどの理由から、まず A1 を行い、つぎに A2 を行ってその回数の和を距離とする。

以上の類似度を基に、免疫産生テストを行う。免疫産生テストは、既存の適応度の大きな解に類似した解を選択するテストであり、以下のように表わされる^{9),10)}。

$$\frac{\sum_{i=1}^{N_p} m_i(i, k) f(x_i)}{\sum_{i=1}^{N_p} m_i(i, k)} \geq \frac{\sum_{i=1}^{N_b} f(x_i)}{N_b} \quad (7)$$

ここで、 N_p は類似性を考慮する解の個数、 N_b は適応度の参照値として利用する解の個数、 $m_i(i, k)$ は二つの解、 x_i, x_k の類似度、 $f(x_i)$ は解の適応度である。右辺はしきい値と呼ばれる。 N_p が大きい場合、計算時間が増大する。また、 N_p と N_b の値の差が大きいほど、テストは厳

しくなり、テストを満足する解がなかなか得られなくなる。このテストにより、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、局所的な探索が活発に行われることになる。

類似度 m_1, m_2 を使用する場合、免疫産生テストは一つの解に対し2回行い、両方のテストを同時に満足する解のみ解集団に残すことにする。二つのテストは完全に独立に行う。

5. 実験結果

実験に使用する問題およびパラメータ値を Table 1, Table 2 に示す。各種パラメータの値は、実験的に決定した。ここで、類似度 m_1, m_2 を使用するものを GIRM1, m_3 を使用するものを GIRM3 とする。また、GIRM1 と GIRM3 をまとめて GIRM と呼ぶことにする。

1 世代で生成される解の個数は $N \cdot C$ である。突然変異は、1 世代当たり $N \cdot M$ 個の解に対して、交叉および解の選択が終了した後に行う。世代数は、比較のために、扱う問題および手法によらず一定の 2000 とした。Fig. 4 に lattice2 における解の収束の例を示す。Fig. 4 より、2000 世代までで解は十分に収束することがわかる。

(2) 式の λ は 1, c_{ij} は i, j 間に配線がある場合は 1, それ以外の場合は 0 である。

実験は初期集団を変えて 100 回行った。Table 3, Table 4 に各手法で得られた最良の解の評価関数値および平均計算時間を示す。また、Fig. 5 から Fig. 8 に解の

Table 1 Problems for simulations

	lattice1	lattice2
Total number of modules	16	25
The shapes of the modules	square	square
The areas of the modules	1	1
Total number of wires	24	40
The minimum value of the cost function ($\lambda = 1$)	40	65

Table 2 The values of the parameters

parameter	sign	The value of the parameter		
		GIRM1	GIRM3	GA
Population size	N	50	50	50
Crossover rate	C	0.8	0.7	0.7
Mutation rate	M	0.3	0.3	0.3
The number of solutions to determine affinity	N_p	6	4	—
The number of solutions to calculate the threshold	N_b	5	2	—

Fig. 4 An example of the convergence of a solution for lattice 2

Table 3 The experimental results for lattice 1

	GIRM1	GIRM3	GA
The value of the cost function	40.0	40.0	40.0
Time (s)	341	148	96

Table 4 The experimental results for lattice 2

	GIRM1	GIRM3	GA
The value of the cost function	85.0	79.5	85.0
Time (s)	551	221	99

Fig. 5 A frequency of solutions for lattice 1 generated by GIRM 1

Fig. 6 A frequency of solutions for lattice 1 generated by GIRM 3

Fig. 7 A frequency of solutions for lattice 2 generated by GIRM 1

Fig. 8 A frequency of solutions for lattice 2 generated by GIRM 3

Fig. 9 A floorplan for lattice 2 generated by GA

Fig. 10 A floorplan for lattice 2 generated by GIRM 3

頻度分布を示す。

Fig. 9, Fig. 10 に GA および GIRM により得られた lattice2 のフロアプランを示す。

通常の GA と局所探索を取り入れた GIRM を比較すると、GIRM で、評価関数値の小さな解がより多く得られている。顕著な差が現れない理由としては、GIRM では、探索の初期段階から類似性に基づいた解の選択を行うため、GA に比べ解の多様性が失われやすく、局所解に

陥りやすいこと、また、大域的探索が十分に行われないことなどが挙げられる。これらを防ぐために、パラメータ N_p, N_b を世代数により変化させる方法¹⁰⁾なども考えられている。

表現型に基づく類似度を用いる GIRM 1 と遺伝子型に基づく類似度を用いる GIRM 3 を比較すると、GIRM 3 で良い結果が得られている。計算速度についても、GIRM 3 の方が速い。GIRM 1 で使用する類似度 m_1, m_2 を求める際には、得られたフロアプランからモジュールの位置やチップの分割領域の大きさを考慮する。一方、GIRM 3 で使用する類似度 m_3 は、チップの分割構造にのみ関係しており、遺伝子コードから直接求められる。 m_3 のほうがより単純に計算できるため、GIRM 3 の方が計算時間は少なくてすむ。

解の探索は、遺伝子コードに基づき行われるので、遺伝子コードの類似性によって解を選択する GIRM 3 のほうが、局所的探索をより活発に行えると考えられる。そのため、GIRM 1 に比べ GIRM 3 のほうが、評価関数値の小さな解がより多く得られる。

6. む す び

GA では多点探索を行うために、探索範囲が広くなり、また局所解に陥りにくいとされている。しかし、局所的探索性能の低さが指摘されている。

IRM は解の類似性に着目し、コスト的に優れた解に類似した解を選択的に残すものである。そのため、解は急速に一点に収束するが、局所解に陥りやすい。

GIRM は GA に IRM の要素を取り入れ、GA の局所的探索性能を向上させた手法である。本論文では、GIRM をフロアプラン設計に適用した。その結果、GA と GIRM では得られる解の頻度分布に多少の違いが見られた。GIRM のほうが良い解に到達する可能性は高いと考えられるが、類似度の定義によっては GA とほぼ同等の解しか得られない。フロアプラン設計に関しては、類似度を遺伝子コードから直接計算した場合のほうが有効であることがわかった。

今回はフロアプランのコード化に逆ポーランド表記を使用した。逆ポーランド表記は、モジュールの相対的な位置を決定するだけで、座標などを考えないという意味では単純である。しかし、GA における交叉などの解の生成方法を考える場合には、数字以外に符号を含むこと、それらの位置に制限があることなどの理由で、扱いがかなり複雑になる。また、類似度を考える場合にも、遺伝子列から直接定義するのは難しい。今後は、遺伝的な処理や類似度の定義を行いやすい遺伝子列を生成する符号化方法についても考えていく。また、GIRM をほかの問

題に適用することも今後の課題である。

最後に日頃有益なる御助言、御討論いただいている、本学倉田は教授に感謝いたします。

参 考 文 献

- 1) D. F. Wong and C. L. Liu: A New Algorithm for Floorplan Design, in Proc. 23rd ACM-IEEE Design Automation Conf., 101/107 (1986)
- 2) J. P. Cohoon, S. U. Hedge, W. N. Martin and D. S. Richards: Distributed Genetic Algorithms for the Floorplan Design Problem, IEEE Trans. CAD-10-4, 483/492 (1991)
- 3) D. E. Goldberg and J. Richardson: Genetic Algorithms with Sharing for Multimodal Function Optimization, in Proc. 2nd Int. Conf. on Genetic Algorithms, 41/49 (1987)
- 4) K. Deb and D. E. Goldberg: An Investigation of Niche and Species Formation in Genetic Function Optimization, in Proc. 3rd Int. Conf. on Genetic Algorithms, 42/50 (1989)
- 5) 小坂, 平田: 組み合わせ最適化問題における生物学的方法—遺伝的アルゴリズムとシミュレーテッドエボリューション—, 電学論 C, **112**-10, 585/590 (1992)
- 6) H. Bersini and F. J. Varela: The Immune Recruitment Mechanism: Selective Evolutionary Strategy, in Proc. 4th Int. Conf. on Genetic Algorithms, 520/526 (1991)
- 7) Robert E. Smith, Stephanie Forrest and Alan S. Perelson: Population Diversity in an Immune System Model: Implications for Genetic Search, in Foundation of Genetic Algorithms 2, 153/165, Morgan Kaufmann (1993)
- 8) J. D. Farmer, N. H. Packard and A. S. Perelson: The Immune System, Adaption, and Machine Learning, Physica 22D, 187/204 (1986)
- 9) 森, 築山, 福田: 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**-10, 872/878 (1993)
- 10) H. Bersini and F. J. Varela: Hints for Adaptive Problem Solving Gleaned from Immune Networks, in Proc. of 1st Workshop of PPSN, 343/354 (1990)
- 11) S. Forrest and A. S. Perelson: Genetic Algorithms and the Immune System, in Proc. of 1st Workshop of PPSN, 321/325 (1990)
- 12) 和田, 田中, 和田: 進化システム論遺伝的アルゴリズムの基礎(6), in Computer Today, **53**-1 (1993)
- 13) 小山, 大沢: 免疫学の基礎, 東京化学同人 (1989)

[著 者 紹 介]

田 澤 功

1995 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。現在, 同大学院自然科学研究科生産科学専攻博士課程に在学中。VLSI レイアウト設計の研究に従事。電子情報通信学会, 電気学会などの会員。

小 坏 成 一 (正会員)

1992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。同年同大学工学部助手となり現在に至る。VLSIレイアウト設計,確率的最適化手法の研究に従事(工学博士)。電子情報通信学会,電気学会などの会員。

平 田 廣 則 (正会員)

1976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,並びに分散システム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,電気学会,情報処理学会,システム制御情報学会などの会員。

=====