

## 適応的遺伝的アルゴリズムとシーケンスペアに基づく フロアプランニング手法

中矢 真吾<sup>†</sup> 若林 真一<sup>†</sup> 小出 哲士<sup>†</sup>

<sup>†</sup>広島大学工学部

〒739-8527 東広島市鏡山一丁目4番1号

<sup>†</sup>東京大学 大規模集積システム設計教育研究センター

〒113-8656 文京区本郷7-3-1

E-mail:{nakaya, wakaba}@ecs.hiroshima-u.ac.jp, koide@vdec.u-tokyo.ac.jp

本稿ではVLSIレイアウト設計におけるフロアプランニング問題に対し、適応的遺伝的アルゴリズムに基づく新しい手法を提案する。提案手法においては、問題の解をシーケンスペアで表現する。さらに、本稿で新たに提案する複数の交差手法を個体ごとに動的に選択することにより、短い計算時間で優良解を求めることを可能にしている。シミュレーテッドアニーリングと非適応的遺伝的アルゴリズムとの比較実験の結果、提案手法の有効性が検証された。

キーワード：フロアプランニング，シーケンスペア，適応的遺伝的アルゴリズム，エリート度，適応的選択

## An Adaptive Genetic Algorithm with Sequence-Pair Representation of Solutions for VLSI Floorplanning

Shingo NAKAYA<sup>†</sup>, Shin'ichi WAKABAYASHI<sup>†</sup> and Tetsushi KOIDE<sup>†</sup>

<sup>†</sup>Faculty of Engineering, Hiroshima University

4-1, Kagamiyama 1 chome, Higashi-Hiroshima 739-8527 JAPAN

<sup>†</sup>VLSI Design and Education Center, The University of Tokyo

7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656 JAPAN

E-mail:{nakaya, wakaba}@ecs.hiroshima-u.ac.jp, koide@vdec.u-tokyo.ac.jp

This paper proposes a new floorplanning algorithm in VLSI layout design, which is based on an adaptive genetic algorithm. In the proposed algorithm, a solution of the problem is represented as a sequence-pair. The proposed algorithm can find a good solution in a short computation time by selecting dynamically one among newly proposed crossover operators to be applied to each individual. Experimental results show the effectiveness of the proposed algorithm compared to simulated annealing and non-adaptive genetic algorithms.

key words : floorplanning, sequence-pair, adaptive genetic algorithm, elite degree, adaptive selection

## 1 はじめに

超大規模集積回路 (VLSI) のレイアウト設計における設計工程の一つに、論理機能を実現するモジュールの集合を与えられた制約の下で、チップ面積が最小となるように配置するフロアプランニングがある。フロアプランニングは、チップの面積最小化とパフォーマンス向上のために重要な工程であるが、配線長、タイミング制約を同時に考慮する必要があるため、最適に解くことは困難である。そのため、シミュレーテッドアニーリング (SA)、遺伝的アルゴリズム (GA) といった確率的アルゴリズムを用いて近似解を求めることが一般的である。これらのアルゴリズムにおける解の表現方法として、従来は配置領域を水平、垂直の線分で再帰的に分割して得られるスライシング構造を2分木で表したスライシング木が用いられてきたが、近年、村田らにより非スライシング構造も表現可能なデータ構造としてシーケンスペア (Sequence-Pair) [8], [9] が提案された。シーケンスペアを用いると、与えられた方形の集合に対し、有限な解空間で最適なパッキングを求めることが可能となる。

一方、GA [2] は、複雑な組合せ最適化問題を解くアルゴリズムとして知られている。GA は自然界の進化モデルに基づいて考案された確率的アルゴリズムであり、問題の許容解を1つの個体とみなし、複数の個体からなる集団 (個体群) に対して、遺伝操作を繰り返し適用することにより解を探索する。しかし GA は優れた解探索能力を持つ一方で、解の探索に大きく影響する交差手法の選択やオペレータ適用確率等のパラメータの設定が難しいという問題点がある。

この GA の問題点を解決するため、著者らは文献 [4], [5] において問題のインスタンスに応じて GA パラメータを自動的に設定する適応的 GA (EAGA) を提案している。EAGA では、エリート度と呼ばれる先祖にどれくらい優れた個体を持つかによって定義される個体の潜在的な優劣度を表す指標を用いる。エリート度により、交差手法、突然変異確率を動的に選択することにより、解空間を効果的に探索し質の良い解を得ることが可能となる。

本稿では、フロアプランニング問題に対し、配置を表現する方法としてシーケンスペアを用いて個体をコーディングし、EAGA を適用する手法を提案する [6]。また、提案手法では、個体の選択にスケールを用いることにより、初期探索では大域的な探索を行い、探索が進むにつれて局所探索を強めるこ

とにより、短い計算時間で効果的な探索を可能としている。ベンチマークデータ、テストデータに対するシミュレーション実験により従来手法である SA による解と従来の GA である非適応的 GA による解との比較を行い、提案手法を実験的に評価する。

本稿の構成は以下の通りである。まず、2. でシーケンスペアと EAGA について述べる。次に 3. で EAGA に基づく提案手法について説明する。4. では計算機上での実験結果を示すとともにその有効性を示す。最後に 5. で本稿のまとめを行い、今後の課題について述べる。

## 2 準備

### 2.1 シーケンスペア

シーケンスペア (方形系列対) は配置の対象となるモジュール名から構成される系列  $(\Gamma_-, \Gamma_+)$  により方形配置を表現する方法である。各モジュール名は  $(\Gamma_-, \Gamma_+)$  のそれぞれの系列において1個ずつ含まれ、 $(\Gamma_-, \Gamma_+)$  の順列に基づいて任意の2つの方形の相対的な位置関係が指定される [8], [9]。

図1にシーケンスペアの概念図を示す。図 (a) は、任意の2つのモジュール  $X, Y$  が与えられたときの上下左右の相対位置とシーケンスペアとの関係を表しており、図 (b) は図 (a) を視覚的に示している。すべてのモジュールに対して図1のような相対位置を決定すれば、垂直、水平方向で互いに干渉せずにパッキングを行うことができる。

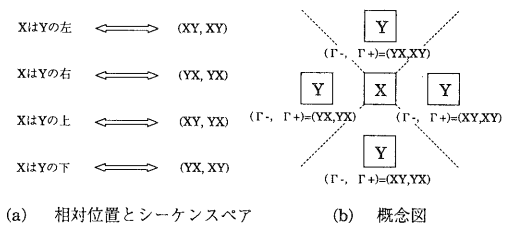


図1 シーケンスペア概念図

シーケンスペアでは、任意の系列対  $(\Gamma_-, \Gamma_+)$  に対してこれらの位置関係を満たす方形パッキングは必ず存在し、しかも、その逆に、任意の方形パッキングに対して系列対  $(\Gamma_-, \Gamma_+)$  が存在する。したがってシーケンスペアを用いることによって非スライシング構造を含む任意の方形パッキングを求めることが可能となる。

提案手法では解 (個体) のコーディング方法としてシーケンスペアを用いる。モジュール数  $n$  の場合、個体の染色体は長さ  $2n$  のシーケンスペア  $(\Gamma_-,$

$\Gamma_+$ ) と長さ  $n$  のモジュールの向きにより構成され、染色体長を  $3n$  としてコーディングする。なお、モジュールの向き  $\theta$  は、端子位置を考慮するため、上下左右、表裏を組合せた 8 通り ( $\theta=0\sim 7$ ) とする。

例えば、図 2(a) の配置は、図 (b) のような相対配置と図 (c) のような個体により表される。

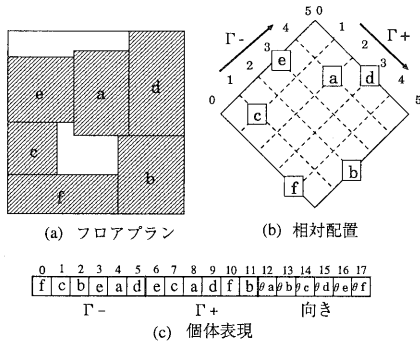


図 2 個体表現の例 (モジュール数 6)

モジュールの相対位置が決定すると、全モジュールの相対位置により作成される水平、垂直制約グラフ (それぞれ  $G_h, G_v$  とする) によりチップ面積を求める (図 3)。  $G_h, G_v$  では、それぞれモジュールの幅、高さが辺の重みとして与えられ、始点から終点までの最長経路を求める。それぞれの最長経路上の辺の重みの総和がチップ幅、チップ高さに相当し、その積によりチップ面積と各モジュール位置を決定する。チップの配置が決まると、各端子の位置も決定するため、各ネットの配線長を半周近似により求め、全ネットの配線長の和をとることによりチップ配線長を決定する。

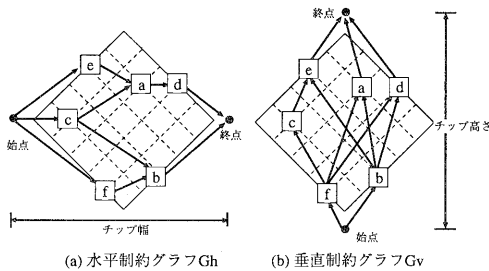


図 3 制約グラフ

## 2.2 評価関数

GA では、交差、突然変異の遺伝的操作を行った後、次世代に残すべき個体を選択するため個体を評価する。評価方法は、シーケンスペアから水平、垂直制約グラフの最長経路長の積よりチップ面積

$A[mm^2]$ , 半周近似によるネットの配線長の総和よりチップの配線長  $W[mm]$  を求める。このとき、個体の評価値  $E(A, W)$  を以下のように定義する。

$$E(A, W) = A + k \times W \quad (1)$$

ここで、 $k$  は配線長の重み定数であり、実験では  $k=0.005$  としている。

## 2.3 適応的遺伝的アルゴリズム

GA 実行中に GA で用いられるパラメータ値を適応的に調整する GA を適応的 GA と呼ぶ。これまでに多くのパラメータ値の適応的な調整手法が提案されているが、それらの大半は個体集合全体に対するパラメータ値の調整のみを行っており、個体ごとに適用するパラメータ値を調整する手法はほとんどない。そこで、著者らは文献 [4], [5] において計算時間の短縮と解の質の向上を目的として、GA 実行中に個々の個体について適用するパラメータ値を動的に調整する適応的遺伝的アルゴリズムを提案している。以下にその概要を示す。

個体の潜在的な優劣の度合を示す指標としてエリート度 (*Elite degree*) を定義する。初期解の世代を世代 0 とし、現在の世代を  $T$  としたとき、世代  $T$  における  $i$  番目の個体を  $x_i^T$  ( $0 \leq i \leq Pop\_size - 1$ ,  $Pop\_size$  は個体数) とし、その評価値を  $f(x_i^T)$  とする。さらに、個体  $x_i^T$  の  $j$  世代前の先祖の集合を  $Anc(x_i^T, j)$  とする。エリート度を定義するため、まずエリートを定義する。最大化問題において、ある個体がある世代においてエリートであるとは、その世代の中の個体の評価値の分布を正規分布と仮定して、評価値の平均値を  $\mu_T$ 、評価値の標準偏差を  $\sigma_T$  としたときに以下の条件を満たすことである。

$$f(x_i^T) \geq \mu_T + \alpha \times \sigma_T \quad (2)$$

ここで、 $\alpha$  は非負の実数で、エリート決定係数と呼ぶ。 $\alpha$  を大きくするとエリートとみなされる個体数が少なくなり、逆に小さくすると多くなる。このとき、世代  $T$  における  $i$  番目の個体  $x_i^T$  のエリート度  $E\_deg(T, i)$  を以下のように定義する。

$$E\_deg(x_i^T) = \frac{\sum_{j=0}^{imax} \{|Elite(x_i^T, j)| \times \beta^j\}}{\sum_{j=0}^{imax} \{|Anc(x_i^T, j)| \times \beta^j\}} \quad (3)$$

ここで、 $Elite(x_i^T, j)$  は個体  $x_i^T$  に対する世代  $T-j$  でのエリートである先祖の集合、 $\mu_T, \sigma_T$  はそれぞれ世代  $T$  における個体の評価値の平均値と標準偏差、 $f(x_i^T)$  は世代  $T$  における  $i$  番目の個体の評価値、 $\alpha$  はエリート決定係数、 $\beta$  ( $0 \leq \beta \leq 1$ ) はエリート影響度係数である。 $\beta$  を変化させることにより、遠い世

代の先祖の影響を減少させることができる。ここでは最大化問題に対するエリート度を定義したが、最小化問題に対するエリート度も同様に定義できる。

エリート度の計算例を図4に示す。図4で節点は個体を表し、枝は個体の親という関係を表す。ハッチのついた節点はエリートの個体を表す。この例で先祖へたどるレベルの最大範囲  $level\_max$  は3である。

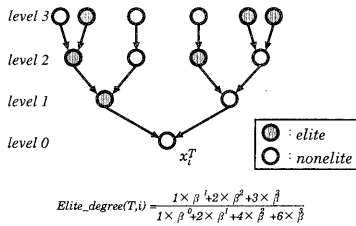


図4 エリート度の計算例

エリート度により個体の潜在的な優劣を判別できるため、解の効果的な探索が可能となる。交差手法において、エリート度の高い個体には親の性質を多く残す交差手法、エリート度の低い個体には解空間を広く探索する交差手法を用いる。突然変異確率についても同様の考えで、エリート度の高い個体には低い突然変異確率、エリート度の低い個体には高い突然変異確率で突然変異を行い、エリート度の高い個体の良い性質を壊さないようにする。

### 3 提案フロアプランニング手法

本節ではフロアプランニング問題に対する、エリート度に基づく適応的遺伝的アルゴリズム (EAGA) を提案する。

#### 3.1 エリート交差手法

本稿ではエリート個体用の交差手法として共通トポロジー保存交差 (Common Topology Preserving Crossover, CTPX) を提案する。この手法は親の共通するモジュールの位置関係をできるだけ多く保存する手法である。具体的には、シーケンスでのモジュールの順序はそれらのモジュールの上下左右の位置関係を示していることから、親のシーケンス ( $\Gamma_-$ ,  $\Gamma_+$ ) の中で共通の順序を持つモジュール列の中で最長のもの (最長共通部分列, Longest Common Subsequence, LCS) を求める。

LCSに含まれるモジュールは両方の親で同じ順序に含まれているため、同じ相対位置を持つことになり、これらのモジュールの順序が親に共通する性質であると言える。LCSに含まれるモジュールは交差の対象から外し、系列内の位置と向きを交差時に

固定する。逆に LCS 以外のモジュールは系列の順序と向きを交差の相手の親の順に変更する。

図5にCTPXの例を示すとともに、以下にアルゴリズムを示す。

#### 【アルゴリズム：CTPX】

- ステップ 1: 2つの親  $p_1, p_2$  の  $\Gamma_-$ ,  $\Gamma_+$  において LCS を求め、LCS に含まれるそれぞれのモジュールの集合を  $M_-$ ,  $M_+$  とする。
- ステップ 2:  $M_-$  のモジュールに対して  $p_1, p_2$  の  $\Gamma_+$  の LCS を求める。そのモジュールの集合を  $M_{-+}$  とする。
- ステップ 3:  $M_+$  のモジュールに対して  $p_1, p_2$  の  $\Gamma_-$  の LCS を求める。そのモジュールの集合を  $M_{+-}$  とする。
- ステップ 4:  $M_{-+}$ ,  $M_{+-}$  で、数の多い方を  $M_{nc}$ 、他方のモジュールを  $M_c$  とする。
- ステップ 5:  $M_{nc}$  のモジュールをそのまま  $p_1, p_2$  から子にコピーする。
- ステップ 6:  $M_c$  のモジュールを  $\Gamma_-$ ,  $\Gamma_+$  で他方の親の順に並び変えて子にコピーする。

図5の例では、 $\Gamma_+$  の LCS は  $(c, a, d, b)$  であり、 $\Gamma_-$  の LCS は  $(c, b, a, d)$  である。そうすると、LCS のモジュール  $(c, a, d, b)$  は、親1、親2とも  $\Gamma_-$  では  $\{c \rightarrow b \rightarrow a \rightarrow d\}$  となり、LCS のモジュール  $(a, b, c, d)$  を得る。逆に  $\Gamma_-$  の LCS  $(c, b, a, d)$  を  $\Gamma_+$  で見ると、親1、親2とも  $\{c \rightarrow a \rightarrow d \rightarrow b\}$  となっているから同様に LCS のモジュール  $(a, b, c, d)$  を得る。この場合、モジュールの個数は同じでかつ同じモジュールであるため、それらのモジュールに対しては位置を固定する。それ以外のモジュール (この例では  $e, f$ ) に対しては、他方の親の順に並び変える。実際のアルゴリズム CTPX ではさらにステップ6において、個体の評価値が高い程、実際に並び変えられるモジュールが少なくなるようにモジュールを選択し、選択されたモジュールだけ順序と向きを他方の親に合わせている。

#### 3.2 非エリート交差手法

非エリート交差手法として、本稿では、配置依存部分交換交差 (Placement-based Partially Exchanging Crossover, PPEX) を提案する。PPEX は、乱数により選択されたモジュールを中心に窓領域を作成し、窓に含まれたモジュールを交差対象とし交差を行う。ここで窓とは、シーケンスペアで定義されるモジュールの相対位置を表す斜め格子 (オブリークグリッド, 図2(b)参照) 上の連続した部分領域を意味する。これにより、配置上、近いモジュールに関

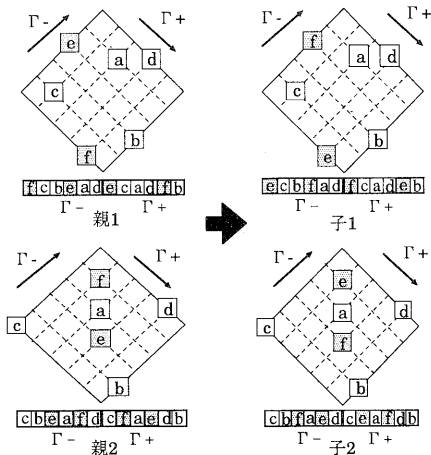


図 5 CTPX

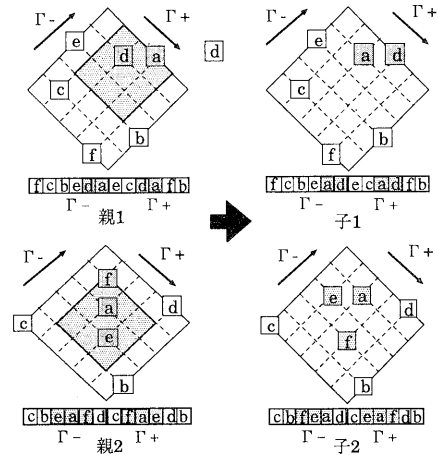


図 6 PPEX

して局所的な交差を行うことが可能となる。アルゴリズムを以下に示す。

【アルゴリズム：PPEX】

- ステップ 1： それぞれの親から乱数によりモジュールを 1 個ずつ選択する。
- ステップ 2： ステップ 1 で選択したモジュールを中心に窓領域を生成し、窓に含まれるモジュール集合を交差対象  $M_c$  とし、それ以外のモジュール集合を  $M_{nc}$  とする。
- ステップ 3：  $M_c$  の各モジュールを  $\Gamma_-$ ,  $\Gamma_+$  で相手の親の順に入れ換え、子にコピーする。
- ステップ 4：  $M_{nc}$  の各モジュールはそのまま子にコピーする。

PPEX で窓領域の 1 辺を 4 としたときの例を図 6 に示す。この例では、親 1 でランダムに生成された窓領域に含まれるモジュール  $a, d$  を交差対象モジュールとする。交差前は  $(\Gamma_-, \Gamma_+) = (da, da)$  であるが、交差後親 2 の順  $(\Gamma_-, \Gamma_+) = (ad, ad)$  に変更している。向きについても同様に相手の親の向きに変更する。親 2 についても同様の操作を行う。

3.3 突然変異 MM

提案手法で用いる突然変異 MM について述べる [6]。以下に MM のアルゴリズムを示す。

【アルゴリズム：突然変異 MM】

- ステップ 1： モジュール  $x$  を突然変異確率に応じて選択する。ここで、 $x$  は  $\Gamma_-$ ,  $\Gamma_+$  の

- 中それぞれ  $i, j$  番目であるとする。
- ステップ 2： ランダムに  $k, l$  ( $1 \leq k, l \leq n$ ) を決定する。

- ステップ 3：  $x$  を  $\Gamma_-$  においては  $k$  番目のモジュールと、 $\Gamma_+$  においては  $l$  番目のモジュールと交換する。
- ステップ 4：  $x$  の向きをランダム (0~7) に変更する。

図 7 に MM の例を示す。モジュール  $a$  が乱数により選ばれ、ランダムに位置変更する。モジュール  $c, d$  は、 $\Gamma_-$ ,  $\Gamma_+$  において  $a$  と重なるので、 $a$  が元にあった位置に変更する。また、本手法ではしきい値により、エリート、非エリートの個体に分け、それぞれ突然変異確率  $P_e, P_{ne}$  ( $P_e < P_{ne}$ ) で MM を適用する。これにより、エリート個体の良い性質を壊さないようにしている。

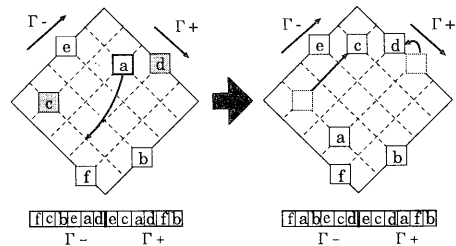


図 7 突然変異 MM

### 3.4 ヒューリスティックによる局所改良

提案手法では、さらなる解の改善のために、局所改良を行う。局所改良のアルゴリズムを以下に示す。  
【アルゴリズム：局所改良】

- ステップ 1： 最長経路上のモジュールを 1 個選択して向きを変えて評価値が最小となる向きに変更する。
- ステップ 2： 最長経路上のモジュールを 1 個選択し、 $\Gamma_-$ 、 $\Gamma_+$  の位置、向きを変更し評価値の改善を行う。評価値が改善されなければ、元の位置と向きのままにする。
- ステップ 3： ステップ 1、ステップ 2 を繰り返し、指定回数解が改善されない場合、又は、指定回数繰り返された場合終了する。

局所改良では、世代数が進むに従い、指定回数を多くすることにより、前半では、局所改良を抑え、後半では、局所改良を強くしている。こうして、効率的に解の改善を行い、回数の指定により局所解に陥らないようにしながら局所改良することを可能としている。

### 3.5 スケーリングによる個体の選択

提案手法では、選択時にスケーリングを用いて個体の評価値を一時的に変更することにより、解の効果的探索を可能としている。スケーリングにより変更された個体の評価値  $f'$  は (4) 式のように表される。

$$f' = a \times f + b \quad (4)$$

この式で、 $a$ 、 $b$  は実数パラメータで世代が進むにつれ変化する。表 1 に終了世代数  $G_{max}$  としたときのパラメータ  $a$ 、 $b$  の値の変化を示す。

表 1 世代数  $G_{max}$  としたときのパラメータ  $a$ 、 $b$  の値

開始世代	終了世代	$a$	$b$
1	$\frac{1}{4}G_{max}$	0	$f_{min}$
$\frac{1}{4}G_{max}+1$	$\frac{3}{4}G_{max}$	$1 - \frac{b}{f_{min}}$	$f_{min}(\frac{3}{2} - \frac{2G}{G_{max}})$
$\frac{3}{4}G_{max}+1$	$G_{max}$	1	0

表 1 で、 $f_{min}$  は面積下界値と最良配線長による評価関数であり、 $G$  は世代数である。スケーリングでは、初期世代で係数  $a = 0.5$  とし、個体間の評価値の差を実際より圧縮することにより、選択時の評価値の悪い個体の淘汰を抑制し、個体に多様性を持たせることができる。そして、探索が進むにつれて  $a$  を徐々に 1 に近づけることにより評価値の悪い個体を淘汰し、より局所探索を強めるようにする。ス

ケーリングにより、短時間で効果的な解の探索が可能となる。

### 3.6 提案手法

提案手法全体のアルゴリズムフローを図 8 に示す。評価、局所改良、選択、交差、突然変異はそれぞれ前節で説明したものをを用いる。主として解の探索は、交差、突然変異、局所改良で行い、得られた個体を評価し、次世代に残す個体を選択する。選択では、個体の評価値に比例した確率で次世代に残す個体を選択するルーレット選択を行う。選択により評価値の悪い個体は淘汰される。

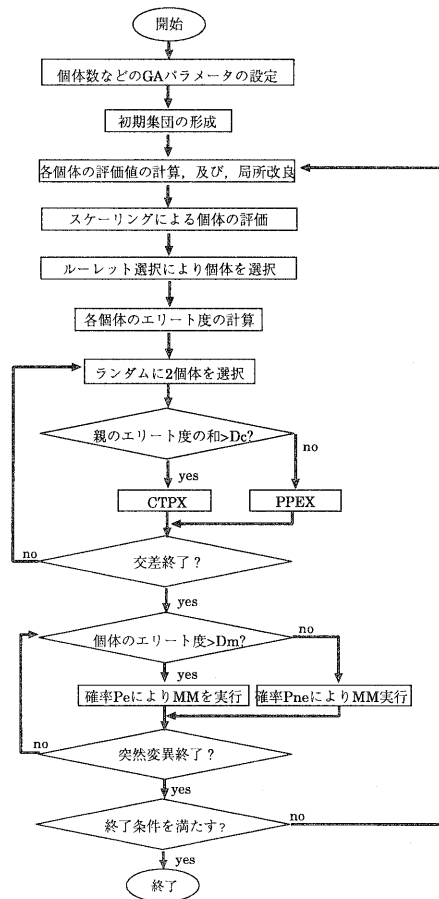


図 8 提案手法全体のアルゴリズム

#### 4 実験的評価

本稿で提案したアルゴリズムを C 言語により実現し、ワークステーション Ultra COMP Station Model 170(167MHz) 上でシミュレーション実験を行った。データは、2 種類の MCNC ベンチマークデータ (ami33, ami49) とモジュール数 100 のテストデータ (rd100) を使用している。本実験の提案手法で用いた GA のパラメータを表 2 に示す。実験では従来手法としてシミュレーテッド・アニーリング (SA) と非適応的 GA を比較対象とし、各手法で得られる評価値、面積、配線長を比較する。実験結果を表 3, 4, 5 に示す。結果は、10 回の実験による最良値、最悪値、平均値を示している。

提案手法の面積の最良値における各データのレイアウト図を図 9, 10, 11 に示す。

##### 4.1 SA と提案手法の比較

従来手法としてシミュレーテッド・アニーリング (SA) を対象とし、シミュレーション実験により得られた評価値を比較する。表 5 から、どのデータに対しても、評価値に関しては最良値、最悪値、平均値において、短時間で良い結果が得られている。しかし、表 3 から面積では提案手法の方が良い結果が得られているが、表 4 から配線長においては SA の方がほとんど良い結果が得られている。これは SA では局所探索が強いためある程度バックキングしてからも配線長の最小化が行えるのに対し、EAGA では交差による大域探索により配置において良いバックキングを求めることができるが、配線長を改善するための局所探索が弱いと考えられる。しかし、評価値、計算時間から見れば、提案手法の方が優れており、提案手法の有効性が確認できる。

##### 4.2 交差を単独で用いた場合との比較

交差手法、突然変異確率を 1 種類しか用いない非適応的 GA と提案手法の結果を比較する。結果から平均値においてはすべての項目で提案手法が同じか良い結果が得られた。交差手法 CTPX では、最良値では提案手法より良い結果が得られるが、最悪値は悪い傾向にある。これはすべての個体に CTPX を適用すると個体に共通する LCS が求められにくいためだと考えられる。また、PPEX では CTPX とは逆に最良値、最悪値に差がなく、安定した値が得られている。以上のことから、CTPX, PPEX は単独に用いるのは不向きで、個体のエリート度による適応的選択を行うと解の質が向上することが分かる。

表 2 提案手法の設定パラメータ

・世代数	2000
・個体数	20
・交差確率	0.6
・エリート突然変異確率	0.01
・非エリート突然変異確率	0.05
・エリート度影響係数 ( $\beta$ )	0.5
・エリート決定係数 ( $\alpha$ )	0.2
・最大局所改良回数	10
・交差手法	適応的交差
・適応的交差手法選択で用いるしきい値	1.5

表 3 面積と計算時間の比較結果

データ	手法	面積 (mm)			計算時間 [sec]
		最良値	最悪値	平均値	
ami33	SA	1.262	1.287	1.276	4148
	GA(CTPX)	1.218	1.311	1.279	784
	GA(PPEX)	1.233	1.278	1.257	720
	提案	1.230	1.268	1.257	811
ami49	SA	39.41	40.22	39.81	7942
	GA(CTPX)	37.85	39.44	38.53	1144
	GA(PPEX)	38.15	38.85	38.62	1032
	提案	37.52	38.92	38.13	1223
rd100	SA	10.134	10.458	10.268	15956
	GA(CTPX)	10.042	10.467	10.185	2040
	GA(PPEX)	10.080	10.445	10.217	1925
	提案	9.785	10.037	9.880	2341

表 4 配線長の比較結果

データ	手法	配線長 (mm)		
		最良値	最悪値	平均値
ami33	SA	41.87	53.72	47.22
	GA(CTPX)	44.92	60.23	52.14
	GA(PPEX)	48.62	54.52	51.94
	提案	42.68	52.15	46.20
ami49	SA	1020.2	1147.8	1087.3
	GA(CTPX)	1101.1	1338.7	1229.3
	GA(PPEX)	1208.3	1391.6	1313.3
	提案	1132.7	1351.4	1205.2
rd100	SA	653.1	678.7	667.7
	GA(CTPX)	707.7	765.9	745.3
	GA(PPEX)	712.1	777.7	737.9
	提案	683.1	719.7	705.0

表 5 評価値の比較実験

データ	手法	評価値		
		最良値	最悪値	平均値
ami33	SA	1.471	1.545	1.512
	GA(CTPX)	1.463	1.587	1.519
	GA(PPEX)	1.471	1.533	1.512
	提案	1.463	1.506	1.488
ami49	SA	44.61	45.76	45.24
	GA(CTPX)	43.38	45.91	44.67
	GA(PPEX)	44.42	45.06	44.69
	提案	44.39	44.90	44.16
rd100	SA	13.49	13.85	13.61
	GA(CTPX)	13.68	14.28	13.91
	GA(PPEX)	13.70	14.11	13.91
	提案	13.21	13.44	13.34

## 5 あとがき

本稿では、VLSI 設計におけるフロアプランニング問題に対し、シーケンスペアにより解をコーディングし、エリート度の概念に基づくパラメータの調整機構を持つ適応的遺伝的アルゴリズムを提案し、計算機上にて実現して評価実験を行った。提案アルゴリズムでは、個体のエリート度を計算し、エリート度の高い個体には局所探索能力の高い交差手法を選択し、エリート度の低い個体には解空間を広く探索する交差手法を選択することにより効果的な解の探索を実現した。さらにスケールングを取り入れることにより、短時間で解の向上を実現した。提案手法を従来手法である SA と比較するシミュレーション実験を行った結果、評価値の平均において MCNC ベンチマークデータ ami33 では 1.6%, ami49 では 2.4%, テストデータでは 2.0% 程 SA より提案手法の方が優れており、計算時間の点からも提案手法の有効性を確認した。また、交差手法、突然変異確率を固定した非適応的 GA と比較し、交差手法の適応的選択の有効性を確認した。

今後の課題としては、大規模データへの適用やタイミング制約などの各種制約を考慮したフロアプランニング手法への拡張が挙げられる。

謝辞: 本研究の一部は、文部省科学研究費補助金 奨励研究 (A) (課題番号 10750248), 及び、基盤研究 (c)(2)(課題番号 10680356) による。

## 参考文献

- [1] L. Davis: "Adapting operator probabilities in genetic algorithms," Proc. 3rd International Conference on Genetic Algorithms, pp. 61-69 (1989).
- [2] D. E. Goldberg: "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Publishing Company (1989).
- [3] J. J. Grefenstette: "Optimization of control parameters for genetic algorithms," IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-16, No.1, pp. 122-128 (1986).
- [4] K.Hatta et al.: "On-the-fly crossover adaptation of genetic algorithms," Proc. Genetic Algorithms in Engineering Systems: Innovations and Applications, pp. 197-202 (1997).
- [5] 八田, 他: "遺伝的アルゴリズムにおける交差手法の適応的選択の一手法," 電子情報通信学会論文誌, Vol.J81-D-I, No.7, pp. 900-909 (1998).
- [6] K. Hatta et al.: "Solving the rectangular packing problem by an adaptive GA based on sequence-pair," Proc. of Asia and South Pacific Design Automation Conference, 6D.10, pp. 181-184(1999).
- [7] 北野: "遺伝的アルゴリズム", 産業図書 (1993).

- [8] H. Murata et al.: "VLSI module placement based on rectangle-packing by the sequence-pair", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.15, No.12, pp. 1518-1524 (1996).
- [9] H. Murata et al.: "VLSI/PCB placement with obstacles based on sequence-pair", Proc. of International Symposium on Physical Design, pp. 26-31 (1997).
- [10] S. Sait et al.: "VLSI Physical Design Automation", IEEE Press (1995).
- [11] 坂和, 田中: "遺伝的アルゴリズム," 朝倉書店 (1995).

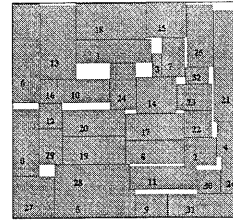


図 9 ami33 の最良レイアウト図 (1.14[mm] × 1.08[mm])

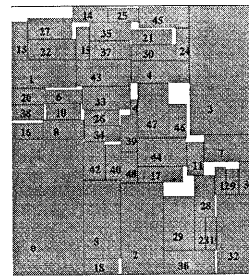


図 10 ami49 の最良レイアウト図 (5.85[mm] × 6.41[mm])

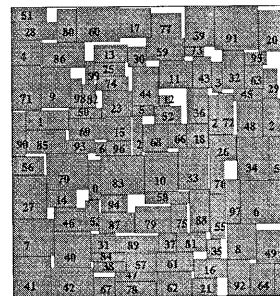


図 11 rd100 の最良レイアウト図 (3.09[mm] × 3.19[mm])