

3次元交叉・突然変異を導入した 3次元FPGA初期配置のための遺伝的アルゴリズム

大村 道郎^{1,a)} 太田 雅也^{2,†1}

受付日 2013年1月4日, 採録日 2013年11月1日

概要: 近年の大規模集積回路における製造技術の進歩にともない, TSV を用いた 3 次元 VLSI 設計が注目を集めている. TSV を使う方法では, 1 つの LSI に 1,000 本以上の TSV を使い配線することができるため, 3 次元に適したアルゴリズムから開発する必要がある. VLSI のレイアウト設計は, 配置設計と配線設計に分かれ, その最初の段階に初期配置設計がある. 本論文では 3 次元 VLSI の中でも特に 3 次元 FPGA 初期配置について議論する. ところで, 一般的に様々な最適化問題に対し, 効率良く準最適解を求めるアルゴリズムの 1 つに, 遺伝的アルゴリズムがあり, マクロセルを含む 3 次元 FPGA の初期配置に対してもそのまま拡張できる. しかし 1 次元の遺伝子列 (個体) で 3 次元の配置を表し, 1 次元配列を対象とした通常の遺伝的的操作を行うと, 配置構造が簡単に壊れて配線長の総和が小さい配置を得られない可能性がある. 本論文では, マクロセルを考慮した 3 次元 FPGA 初期配置問題に対し, 遺伝子に 3 次元構造を持たせ, 3 次元交叉, 3 次元突然変異を導入した, 3 次元 FPGA 初期配置のための遺伝的アルゴリズムを提案する. 実験の結果, 提案手法では立方体の配置領域に対し, 順序交叉, 交換突然変異を単純に適用した場合に比べ, 平均 24.4%配線長の総和が小さい配置を高速に得ることができた. また配置領域を直方体とした場合でも, 従来手法に対し 19.3%配線長の総和が小さい配置を高速に得ることができた.

キーワード: 遺伝的アルゴリズム, 3 次元 FPGA, 初期配置

Genetic Algorithm for 3-D FPGA Initial Placement with 3-D Crossover and Mutation

MICHIROH OHMURA^{1,a)} MASAYA OHTA^{2,†1}

Received: January 4, 2013, Accepted: November 1, 2013

Abstract: As manufacturing technology has advanced in recent years, 3-D VLSI with TSV has been the focus of attention. Because more than 1,000 TSV can be used to design a 3-D VLSI, we need 3-D layout algorithms. VLSI layout design consists of placement and routing, and initial placement is the first stage of this placement. In this paper, initial placement of 3-D FPGA, which is a kind of 3-D VLSI, is discussed. In general, a genetic algorithm is one of the most effective methods for a variety of optimization problem, and it can be applied to the 3-D FPGA initial placement problem with macro cell. Usually, the chromosome is represented by a one dimensional array, and if a 3-D placement region is represented by this chromosome, relative positions of modules are easily broken by the crossover. In this paper, we propose a genetic algorithm with 3-D crossover and mutation, in which the chromosome has 3-dimensional structure. The experimental results show that our approach achieves 24.4% shorter wire length for cube placement regions, and 19.3% shorter wire length for rectangular solid placement regions. In this paper, our proposed method and the experimental results are shown.

Keywords: genetic algorithm, 3-D FPGA, initial placement

¹ 広島工業大学工学部
Faculty of Engineering, Hiroshima Institute of Technology,
Hiroshima 731-5193, Japan
² 広島工業大学大学院工学系研究科
Graduate School of Engineering, Hiroshima Institute of
Technology, Hiroshima 731-5193, Japan
^{†1} 現在, 西日本旅客鉄道株式会社
Presently with West Japan Railway Company
^{a)} ohmura@cc.it-hiroshima.ac.jp

1. はじめに

近年の大規模集積回路における製造技術の進歩にともない, 複数のダイを張り合わせるにより 3 次元に積層し, スルーシリコンビア (TSV) を用いてダイの間を接続する方法を中心に, 3 次元 VLSI 設計に関する研究 [1] が行われるようになってきている. VLSI を 3 次元化すると, パッ

ケーサイズが減少し、配線長が短くなるため、高性能な回路が実現できる。

VLSI レイアウト設計に関しては、通常の 2 次元 VLSI のための EDA ツールは完備されているので、これまでも複数のパッケージを積み重ねて接続し、3 次元化する取り組みがあった [2], [3]。しかし、TSV を使う方法では、1 つの LSI に 1,000 本以上の TSV を使い配線することができるため、これらは通常の 2 次元の EDA ツールをそのまま利用して設計することは不可能であり、3 次元に適したアルゴリズムから開発する必要がある。

VLSI レイアウト設計には、特定の機能を持つ部分回路であるモジュールの配置を行う配置設計と、モジュール間の接続要求を表すネットの配線を行う配線設計がある。配置設計には、チップ上でのモジュールの大まかな配置を求める初期配置と、詳細な配置座標を求める詳細配置がある [4]。3 次元の詳細配置に関して、文献 [5] では、セル間の重なり除去とデバイス層の割当てを同時に行うことにより、配線長とビアの数を最小化しており、また、文献 [6] では、TSV の位置まで決定しているが、本論文では 3 次元 VLSI の中でも特に 3 次元 FPGA の初期配置について議論する。

これまで、2 次元の初期配置 [7] に対し、回路と配置領域の分割、割当てを繰り返すミニカット配置手法 [8] や、2 分割の代わりに 4 分割を行う手法 [9] が提案されている。また、配線長の 2 乗を最小化することにより、長い配線長を持つネットによりペナルティを与える手法が提案されており、この手法は文献 [10] で 3 次元化されている。しかし、これらの手法は、マクロセルを含む 3 次元初期配置には直接対応していない。また、3 次元初期配置については主に 3 次元 FPGA を対象として、モジュール間の張力を用いてグリッドのない配置を求めたのちグリッドに割り付ける手法 [11] や、力学的モデルに基づく手法 [12] が提案されている。しかしこれらの手法もマクロセルは直接考慮していない。

一般的に様々な最適化問題に対し、効率良く準最適解を求めるアルゴリズムの一つに、遺伝的アルゴリズムがある。これは自然界の生物の進化を模倣したアルゴリズムで、解の候補となる、遺伝子で表された複数の個体に対し、選択、交叉、突然変異を繰り返し、適応度の高い個体の集団を求める。これまで、多くの LSI 設計への適用が報告されている [13], [14]。

マクロセルを含む 2 次元 FPGA の初期配置に対してもこの遺伝的アルゴリズムを適用可能で、3 次元 FPGA 初期配置にもそのまま拡張できる。しかし 1 次元の遺伝子列 (個体) で 3 次元の配置を表し、1 次元配列を対象とした通常の遺伝的操作を行うと、たとえ接続の強いモジュールのグループが 3 次元的に近くに配置されていたとしても、配置構造が簡単に壊れて配線長の総和が小さい配置を得られない可能性がある。

本研究では、マクロセルを考慮した 3 次元 FPGA 初期

配置問題に対し、遺伝子に 3 次元構造を持たせ、3 次元交叉、3 次元突然変異を導入した、3 次元 FPGA 初期配置のための遺伝的アルゴリズムを提案する。

実験の結果、まず立方体の配置領域の場合、遺伝的アルゴリズムにおいて順序交叉、交換突然変異を単純に適用した場合に比べ、提案手法では、平均 24.4%配線長の総和が小さい配置を、高速に得ることができた。また、配置領域を直方体とした場合でも、小さいデータに対し、配線長の総和の最適解と比較した場合、従来法では平均 11.9%増加したのに対し、提案手法では平均 6.3%の増加に抑えることができた。さらに、従来手法に対し、19.3%配線長の総和が小さい配置を、高速に得ることができた。本論文では、提案する 3 次元 FPGA 初期配置のための遺伝的アルゴリズムと性能評価のために行った実験結果について述べる。

2. 準備

準備として、まず 3 次元 FPGA について述べ、次に本論文で問題の入力とするモジュールとネットリストについて説明する。

2.1 3 次元 FPGA

3 次元 FPGA では、一般に複数の LSI チップ (ダイ) を張り合わせることで 3 次元に積層し、シリコン貫通電極 (TSV) を用いて LSI チップ相互間の回路接続を行う。図 1 に 3 次元 FPGA [1] における TSV の例 [15] を示す。この例ではダイが 3 枚積み重なっており、それらをつなぐ Z 方向の配線 (図 1 ① 参照) が TSV を表す。

一般に FPGA では、プログラムすることにより、自由に回路を構成できる。図 1 において ② は論理ブロック、③ はスイッチボックス、④ はコネクションボックスを表す。3 次元 FPGA では同じ大きさの論理ブロックが立体的に繰り返し配置されており、本論文でも文献 [11], [12] と同様に、3 次元 VLSI の中でも特に 3 次元 FPGA を対象とする。ハードマクロを含む場合、3 次元 FPGA の配置配線時にはすでにチップに組み込まれており、物理的な位置が与えられる。

2.2 モジュールとネットリスト

チップ上に実現される回路は、それを構成する基本的な

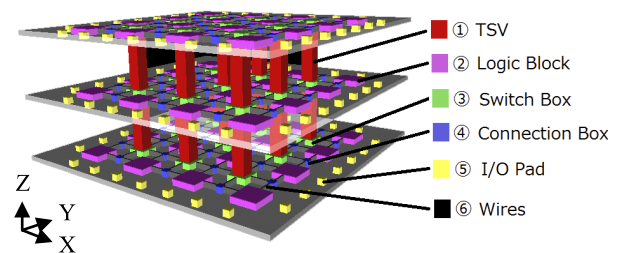


図 1 3 次元 FPGA における TSV
Fig. 1 TSV in 3D FPGA.

機能単位であるモジュールの集合 $M = \{m_i \mid 1 \leq i \leq p\}$ と、モジュール間の配線に関する結線要求を表すネットの集合（以降、ネットリストと呼ぶ） $N = \{n_j \mid 1 \leq j \leq q\}$ によって与えられる。

本論文では、各モジュール $m_i \in M$ は同一の大きさを持つ立方体であり、その中心にネットのための仮想端子を持つと仮定する。各ネット $n_j \in N$ は仮想端子の集合で表されるが、ここでは複雑な表記を避けるため、端子の代わりに単にモジュール名で表す。この場合、同じモジュール m_i に位置する仮想端子はすべて m_i と表されるが、異なるネットに含まれる仮想端子は別の端子と見なすことで、以降の議論に支障はない。

[例 1] 例として、ネット n_1 が m_2, m_3, m_5 の仮想端子を接続し、ネット n_7 が m_1, m_3, m_4, m_8 の仮想端子を接続する場合、 $n_1 = \{m_2, m_3, m_5\}$, $n_7 = \{m_1, m_3, m_4, m_8\}$ と表す。このとき、ネット n_1 の m_3 と、ネット n_7 の m_3 は、同じモジュール m_3 に属する別の仮想端子で、相互につながっていない。 □

3. 3次元初期配置問題の定式化

3.1 3次元配置領域

モジュールを配置する1辺の大きさが u である立方体の3次元領域を3次元配置領域 R と呼ぶ（図 2 参照）。この形状は後で直方体に拡張する。 R は立方格子状に区切られ、それぞれの領域をスロットと呼ぶ。ここでは便宜上、スロットの1辺の長さを1とするが、以降の議論に支障はない。

x 座標、 y 座標、および z 座標が最小であるスロットの中心を座標の原点 $(0, 0, 0)$ とし、その中心の座標が (x, y, z) であるスロットを $sl_{x,y,z}$ で表す（ただし $0 \leq x, y, z \leq u-1$ ）。各モジュールの配置座標、および仮想端子の位置はモジュール m_i が配置されるスロットの中心座標で表すものとする。

本論文では、配置に関し、2種類のモジュールの集合 $M = MC \cup MF$ が与えられるものとする。ここで MC は配置位置が固定され、直方体のマクロセルを形成するモジュールの集合で、各モジュールはそれぞれ個別に仮想端

子を持つものとする。また、 MF は配置位置が固定されていない通常のモジュールであるとする。

3.2 3次元配置問題（立方体）

まず、3次元配置領域 R における各ネットの仮想配線長について説明する。各ネット $n_j \in N$ に接続する仮想端子を囲む最小の直方体のうち、最大座標を (xl_j, yl_j, zl_j) 、最小座標を (xs_j, ys_j, zs_j) とするとき、ネット n_j に対する仮想配線長 l_j を、 $l_j = (xl_j - xs_j) + (yl_j - ys_j) + k \cdot (zl_j - zs_j)$ と定義する。

図 1 において層を垂直に通過するネットは、同一層内のネットに比べて配線にコストがかかると考えられる。仮想配線長 l_j の式において、 z 方向の配線長に定数 $k (\geq 1)$ を掛けることにより、このコストを考慮している。

ネットリスト N に含まれるすべてのネットの仮想配線長の総和 L は、したがって次の式で表される。

$$L = \sum_{n_j \in N} ((xl_j - xs_j) + (yl_j - ys_j) + k \cdot (zl_j - zs_j))$$

以下に3次元初期配置問題を定式化する。

[問題 3DC] 入力として、(1) モジュールの集合 $M = MC \cup MF$ (MC : 配置位置が固定され、直方体のマクロセルを形成するモジュールの集合、 MF : 配置位置が固定されていない通常のモジュール)、(2) ネットリスト N 、(3) 1辺の大きさが u である立方体の3次元配置領域 R が与えられる。このとき、目的関数として配線長の総和 L を最小化する、各モジュールの、 R 内のスロット $sl_{0,0,0} \sim sl_{u-1,u-1,u-1}$ への重複のない配置を求めよ（モジュールの総数がスロットの総数より少ない場合、空きスロットがあってもよい）。 □

4. 提案手法

3次元初期配置問題に適用できそうな手法として、反復改良法であるペア交換法、重心緩和法等 [16] が提案されている。しかしこれらの手法は初期解に大きく依存し、局所解に陥ってしまう危険性がある。それに対し、遺伝的アルゴリズムでは、探索のプロセスにおいて解の候補を複数用意し、それらを集団として改善していく多点探索プロセスを用いている。そのため広域的な探索能力に優れ、局所解に捕らわれることなく最適解に達する可能性が高く [16]、様々な最適化問題に適用されている [17]。そこで提案手法では、遺伝的アルゴリズムを取り入れることにした。

一般に、遺伝的アルゴリズムは1次元配列で表現された遺伝子を対象とした遺伝的操作になっており、3次元構造を持った本論文で扱うような3次元配置問題にそれらの遺伝的操作をそのまま適用すると、3次元の配置構造が壊れて配線長の総和が小さい配置を得られない可能性がある。本研究では、マクロセルを考慮した3次元FPGA初期配置

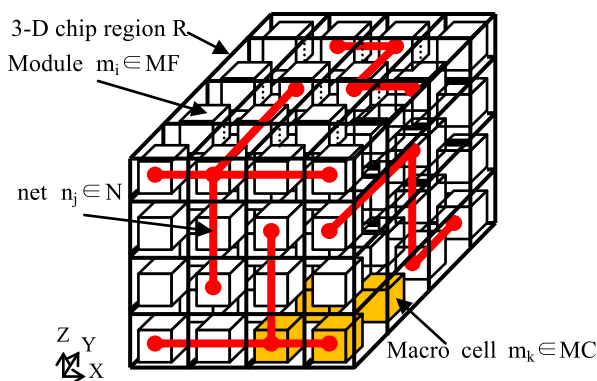


図 2 3次元チップ領域 R
Fig. 2 3-D chip region R .

問題に対し、遺伝子に3次元構造を持たせ、新たなGAオペレータとして3次元交叉と3次元突然変異を導入する。

本論文では、各遺伝子座が各スロットを、各遺伝子はそのスロットに配置されるモジュール番号を表し、各個体が3次元配置領域 R 内へのモジュールの1つの配置を表す。また、モジュールの総数がスロットの総数より少ない場合は、どのネットリストにも接続されていないダミーのモジュールを追加し数を合わせることで、遺伝子の数と遺伝子座の数を一致させる。

遺伝的アルゴリズムでは、与えられた ① モジュールの集合、② ネットリスト、③ 3次元配置領域 R に対し、初期集団として、 R 内のモジュール配置をランダムに複数生成する。このとき、すべての個体でマクロセルの配置位置は共通で固定されている。

4.1 3次元交叉

3次元配置領域 R において、任意の2つのスロットに対し、これらを囲む最小の直方体のうち、マクロセルが含まれる場合はそれらを除いた領域を3次元選択領域 S と呼ぶ。マクロセルが存在する場合、 S は直方体にならない場合がある。この S に含まれるモジュールが次に説明する、3次元交叉の対象となる。

[例2] 例として、図3における2つのスロット $sl_{3,1,1}$ と $sl_{3,3,3}$ を考える。これらのスロットを囲む最小の直方体は、この図の黄色で示すマクロセルを含んでいない。3次元選択領域 S に含まれるモジュールを緑色で示す。

2つの親 A, B が与えられたとき、提案する3次元交叉では、2つの子 C, D を生成する。以下の[手続き3DCO]では、子 C の生成について説明するが、子 D についても同様に生成する。

[手続き3DCO]

(マクロセルがある場合)

S1: R 上のランダムに選択した2つのスロットに対し、これらを囲む最小の直方体のうちマクロセルを除いた3次元選択領域を求める。

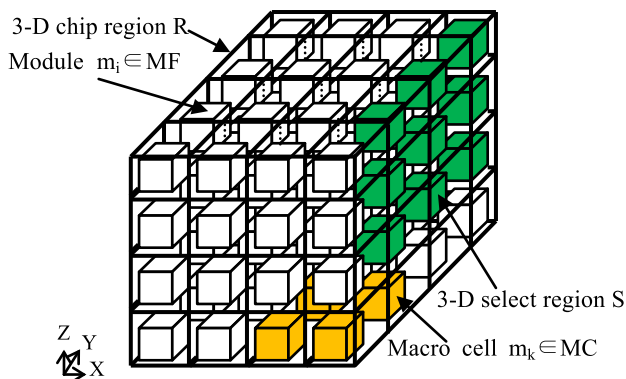


図3 3次元選択領域 S
Fig. 3 3-D select region S .

S2: マクロセル $m_i \in MC$ の遺伝子を A から C に引き継ぐ。

S3: B の3次元選択領域に含まれる遺伝子を C に引き継ぐ。

S4: A のマクロセルと3次元選択領域を除く遺伝子のうち、すでに引き継がれた遺伝子と重複するものを除き、C に引き継ぐ。

(マクロセルがない場合)

S1: R 上のランダムに選択した2つのスロットに対し、これらを囲む最小の直方体となる3次元選択領域を求める。

S2: B の3次元選択領域に含まれる遺伝子を C に引き継ぐ。

S3: A の3次元選択領域を除く遺伝子のうち、すでに引き継がれた遺伝子と重複するものを除き、C に引き継ぐ。

ここで、マクロセルがある場合の S4、またはマクロセルがない場合の S3 において重複した遺伝子(以降、重複遺伝子と呼ぶ)は引き継がないので、必要な遺伝子が欠損(以降、欠損遺伝子と呼ぶ)する場合がある。そのような場合には、該当する遺伝子座(スロット)については空にしておき、次の4.2節で説明する処理によって、仮想配線長が短くなるように、欠損遺伝子の遺伝子座を決定する。

[例3] 例として、モジュール数8、大きさが $2 \times 2 \times 2$ の3次元配置領域を持つ2つの親 A, B が図4(a), (b)のよ

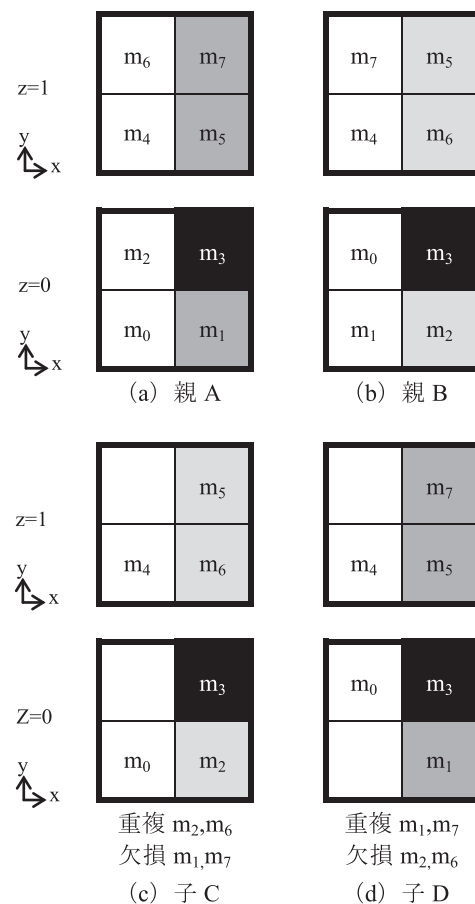


図4 3次元交叉
Fig. 4 3-D crossover.

うに与えられたとする。ここで、黒い正方形はマクロセルを表し、濃い灰色と薄い灰色の正方形は、それぞれ親 A, B の 3 次元選択領域を表す。

まず、マクロセルである m_3 をそのまま子 C, D (図 4(c), (d) 参照) のスロット $sl_{1,1,0}$ に引き継ぐ。次に 3 次元選択領域として、親 B の m_2, m_5, m_6 を子 C に、親 A の m_1, m_5, m_7 を子 D に引き継ぐ。さらに親 A の残り m_0, m_2, m_4, m_6 のうち、すでに子 C に引き継がれている重複遺伝子 m_2, m_6 を除く m_0, m_4 を、親 B の残り m_0, m_1, m_4, m_7 のうち、すでに子 D に引き継がれている重複遺伝子 m_1, m_7 を除く m_0, m_4 を、それぞれ引き継ぐ。子 C の欠損遺伝子は m_1, m_7 、子 D の欠損遺伝子は m_2, m_6 となり、この段階では、子 C のスロット $sl_{0,1,0}, sl_{0,1,1}$ 、および子 D のスロット $sl_{0,0,0}, sl_{0,1,1}$ に対応する遺伝子座はあけておく。 □

4.2 重複、欠損遺伝子の処理

3 次元交叉において、親 B の 3 次元選択領域に含まれる遺伝子の子 C に引き継いだ後、親 A の遺伝子を引き継ぐ際、1 次元の順序交叉における「順序を保存する」に対応する操作、たとえばそれぞれの遺伝子が親 A の元の位置に近くなるよう少しずつずらすような方法は難しく、これまで提案もされていないと考えられるため、ここでは親 A の元の位置と同じ位置に配置できるものを最大限配置し、配置できなかった遺伝子についてはあきらめる（欠損した遺伝子のみにしわ寄せが行く）方法をとる。これにより、多くの遺伝子では、親の配置位置を引き継いでいるものと考えることができる。

欠損遺伝子を空となっている遺伝子座に戻す際、ここでは配線長が短くなる位置に戻す方法を提案する。これは標準的な遺伝的アルゴリズムにはない考え方である。通常の遺伝的アルゴリズムの範囲内と考えられる最も簡単な方法、欠損遺伝子をランダムに戻す方法、との比較により、良い解が得られるということで、本論文では、以下の手法を取り入れることにする。比較結果は 5.1 節で示す。

以下の [手続き RPL] では、4.1 節で空となっている遺伝子座に欠損遺伝子を配置する方法について述べる。ただし、4.1 節で欠損遺伝子の集合を MB 、空となっている遺伝子座の集合を SB とする。

[手続き RPL]

- S1: 任意に 1 つの欠損遺伝子 $m_i \in MB$ を選び、そのモジュールに接続するネットリスト NB_i を求める。
- S2: 空となっているすべての遺伝子座 $sl_{x,y,z} \in SB$ に m_i を入れ、 $n_j \in NB$ の配線長の総和を計算する。
- S3: 配線長の総和が最も短くなる遺伝子座 $sl_{x,y,z} \in SB$ を選び、その遺伝子 m_i を配置する。
- S4: MB から m_i を、 SB から $sl_{x,y,z}$ を除く。
- S5: MB が空でなければ、S1 に戻る。

以下に [手続き RPL] の例を示す。

[例 4] 例として、図 4(c) の子 C における欠損遺伝子を空となっている遺伝子座に配置する場合を考える。欠損遺伝子の集合、空の遺伝子座の集合をそれぞれ $MB = \{m_1, m_7\}$ 、 $SB = \{sl_{0,1,0}, sl_{0,1,1}\}$ とし、 m_1 に接続するネットリストを $NB_1 = \{n_1, n_2\}$ 、 $n_1 = \{m_1, m_5\}$ 、 $n_2 = \{m_1, m_4\}$ 、 m_7 に接続するネットリストを $NB_7 = \{n_3\}$ 、 $n_3 = \{m_2, m_7\}$ とする。ただし、配線長における係数は、 $k = 1$ とする。

まず欠損遺伝子 m_1 をスロット $sl_{0,1,0}, sl_{0,1,1}$ に配置した場合、それぞれの配線長は $(1+0+1) + (0+1+1) = 4$ 、 $(1+0+0) + (0+1+0) = 2$ となり、 m_1 はスロット $sl_{0,1,1}$ に配置する。したがって、 $sl_{0,1,0}$ には残った m_7 を配置する。 □

4.3 3 次元突然変異

遺伝的アルゴリズムにおける突然変異では、遺伝子をランダムに変えることに意味があるが、ここでは、3 次元交叉によって得ることのできない配置を、単体の個体で、しかも遺伝子の 3 次元構造を考慮して変化させることを考える。

1 辺の大きさが u である立方体の 3 次元配置領域において、領域内の xy 平面に平行な面のうち、3 次元配置領域を $0 \leq z \leq i$ と $i+1 \leq z \leq u-1$ ($u-1$ は z の最大値) の 2 つの領域に分ける平面を、(立方体の 3 次元配置領域における) z 分割面と呼ぶ。 x, y 分割面も同様に定義する。

以下の [手続き 3DMT] に、3 次元突然変異の概要を示す。

[手続き 3DMT]

- S1: 3 次元配置領域に対し、 x, y または z 分割面を 1 つ選び、3 次元配置領域を 2 つの領域に分割する。
- S2: 選択された分割面がマクロセルを分割する場合、S1 に戻って選択し直す。
- S3: 分割された片方の領域を、分割面に沿ってランダムに 90 度、180 度、270 度回転をさせる。

4.4 提案手法の概要

3 次元交叉と 3 次元突然変異を導入した遺伝的アルゴリズムの概要を示す。

[アルゴリズム 3DCA]

(S0: 必要があればダミーのモジュールを追加し、遺伝子の数と遺伝子座の数を一致させる。)

- S1: 初期集団をランダムに生成する。
 - S2: 与えられた世代数に達するまで、以下の S2.1–S2.4 を繰り返す。
 - S2.1: 各個体の適応度を求める。
 - S2.2: 適応度により、次世代の個体を選択する。
 - S2.3: 3 次元交叉により、新しい個体を生成する。
 - S2.4: ある確率で 3 次元突然変異させる。
- まず、S1 では、マクロセルの位置を固定し、その他のモ

ジュールの位置をランダムに入れ替えて、初期集団を構成する。次に S2 では、繰り返し回数があらかじめ与えられた世代数 (500 世代) に達すれば終了する。S2.1 の適応度には、配線長の総和の逆数を用いる。S2.2 では、トーナメント選択を用いる。S2.3, S2.4 は、4.1~4.3 節で述べた 3 次元交叉、3 次元突然変異の手法を用いる。

5. 実験結果

提案する遺伝的アルゴリズムを 64 ビット版 CentOS 6.0 上で C++ (gcc version 4.4.6) を用いて実現した。実験は、Intel Core i7 (X 990@3.47 GHz), 12 GByte メモリの計算機上で行った。実験において、遺伝的アルゴリズムのパラメータは、個体数 100, 世代数 500, 交叉率 0.9, 突然変異率 0.05 に設定した。また、配線長における z 方向の定数 k はすべて 1 とした。実験に用いたデータは、すべて乱数を用いて作成した。実際の回路のデータとは異なるが、提案手法の優位性を確認することは可能であると考えられる。

5.1 3次元交叉の有効性

遺伝的アルゴリズムで、遺伝子に重複、欠損が生じない交叉方法として一般に知られている方法に、順序交叉がある。ここでは、3次元交叉の有効性を示すために、3次元の配置を単純に 1次元の遺伝子列 (個体) で表し、順序交叉を用いた従来手法と比較する。ここで、突然変異手法として、両手法とも、ランダムに 2つの遺伝子座を入れ替える交換突然変異を用いる。

$2 \times 2 \times 1$ の大きさのマクロセルを持つ、 $3 \times 3 \times 3$ から $10 \times 10 \times 10$ までの 8種類の立方体データ (C27, C64, ..., C1000) を、乱数を用いて作成した。用意した立方体データの各データ名、モジュールの配置サイズ、ネット数を表 1 に示す。C27~C729 については、各ネットはランダムに 2~5 端子を接続し、C1000 については、各ネットはランダムに 2~10 端子を接続している。またすべてのモジュールはいずれかのネットに接続している。

この 8種類のデータに対し、提案手法と比較手法をそれ

表 1 立方体データ

Table 1 Data for cube region.

Data	Size (X×Y×Z)	Number of nets
C27	3×3×3	50
C64	4×4×4	100
C125	5×5×5	200
C216	6×6×6	300
C343	7×7×7	400
C512	8×8×8	600
C729	9×9×9	800
C1000	10×10×10	1000

ぞれ 10 回ずつ適用し、平均をとった。配線長の総和、CPU 時間、それらの改善率の比較を表 2 に示す。

実験の結果、総配線長で平均 21.6%、提案手法の方が短く、CPU 時間においても平均 41.5%、提案手法の方が短い時間で求めることができている。3次元交叉の有効性が確認された。

データが大きい場合、提案手法が従来手法に比べて非常に速くなっているのは、全モジュール数に対して、交叉に関わる 3次元選択領域に含まれるモジュールの数が、従来手法に比べて圧倒的に少なくなっており、重複遺伝子のチェックにかかる時間が短くなっているのが理由であると考えられる。提案手法では、小さい交叉領域でも、配線長に関して十分良い結果を得ることができている。

次に、空となっている遺伝子座に欠損遺伝子を配置する方法として、4.2 節で述べた提案手法の有効性を示す。提案手法以外に、ランダムに配置する手法が考えられる。これを従来手法とし、提案手法と比較する。ここでは両方の手法に対し、3次元交叉、交換突然変異を適用するものとする。実験結果を表 3 に示す。

実験の結果、提案手法では、従来手法に比べて配線長が平均 5.3%短い結果が得られ、CPU 時間においては平均

表 2 3次元交叉

Table 2 3-D crossover.

Data	Conventional method		Proposed method		Improved ratio [%]	
	L	Time	L	Time	L	Time
C27	185.8	4.6	164.4	9.0	11.5	-95.7
C64	542.4	8.6	417.5	10.2	23.0	-18.6
C125	1425.4	18.7	1105.4	12.1	22.4	35.3
C216	2609.3	44.5	1962.7	15.2	24.8	65.8
C343	4131.3	102.4	3120.5	21.8	24.5	78.7
C512	7135.3	221.8	5416.2	31.4	24.1	85.8
C729	10922.5	444.3	8345.3	47.4	23.6	89.3
C1000	15055.7	813.0	12160.2	70.4	19.2	91.3

表 3 欠損遺伝子の再配置

Table 3 Replacement of gene.

Data	Conventional method		Proposed method		Improved ratio [%]	
	L	Time	L	Time	L	Time
C27	166.3	9.2	164.4	9.0	1.1	2.2
C64	452.2	10.0	417.5	10.2	1.8	-2.0
C125	1132.6	12.0	1105.4	12.1	2.4	-0.8
C216	2027.6	15.6	1962.7	15.2	3.2	2.6
C343	3311.9	20.8	3120.5	21.8	5.8	-4.8
C512	5874.4	30.0	5416.2	31.4	7.8	-4.7
C729	9303.4	44.8	8345.3	47.4	10.3	-5.8
C1000	13500.7	64.1	12160.2	70.4	9.9	-9.8

2.9%時間がかかっているが、ほぼ従来手法と同程度の計算時間で求めることができている。これらの実験結果により、提案する、欠損遺伝子を配置する手法の有効性が確認できた。

5.2 3次元突然変異の有効性

同一のデータに対し、提案手法の3次元突然変異と、一般的な交換突然変異とを比較する。ここで交叉は両手法とも3次元交叉を用いる。提案手法と比較手法をそれぞれ10回ずつ適用し、配線長の総和とCPU時間を比較した。実験結果を表4に示す。

実験の結果、総配線長で平均3.7%、提案手法の方が短く、CPU時間においては5.8%増と、従来手法と比べて少し時間がかかっている。良い解を得るという観点で3次元突然変異の有効性が確認された。

5.3 提案手法の有効性

同一のデータに対し、提案手法として3次元交叉、3次元突然変異を、従来手法として1次元配列で遺伝子を表した順序交叉、交換突然変異を用い、これらの手法を比較する。

提案手法と比較手法をそれぞれ10回ずつ適用し、配線長の総和とCPU時間を比較した実験結果を表5に示す。

表4 3次元突然変異
Table 4 3-D mutation.

Data	Conventional method		Proposed method		Improved ratio [%]	
	L	Time	L	Time	L	Time
C27	164.4	9.0	166.1	9.2	-1.0	-2.2
C64	417.5	10.2	412.7	10.3	1.1	-1.0
C125	1105.4	12.1	1078.3	12.8	2.5	-5.8
C216	1962.7	15.2	1850.6	16.4	5.7	-7.9
C343	3120.5	21.8	2977.9	22.6	4.6	-3.7
C512	5416.2	31.4	5109.5	33.8	5.7	-7.6
C729	8345.3	47.4	7999.8	50.2	4.1	-5.9
C1000	12160.2	70.4	11361.6	79.0	6.6	-12.2

表5 提案手法 (立方体の場合)
Table 5 Proposed method (In the case of cubes).

Data	Conventional method		Proposed method		Improved ratio [%]	
	L	Time	L	Time	L	Time
C27	185.8	4.6	166.1	9.2	10.6	-100.0
C64	542.4	8.6	412.7	10.3	23.9	-19.8
C125	1425.4	18.7	1078.3	12.8	24.4	31.6
C216	2609.3	44.5	1850.6	16.4	29.1	63.1
C343	4131.3	102.4	2977.9	22.6	27.9	77.9
C512	7135.3	221.8	5109.5	33.8	28.4	84.8
C729	10922.5	444.3	7999.8	50.2	26.8	88.7
C1000	15055.7	813.0	11361.6	79.0	24.5	90.3

実験の結果、総配線長で平均24.4%、提案手法の方が短く、CPU時間においては39.6%、提案手法の方が短い時間で求めることができおり、3次元交叉、3次元突然変異を用いた提案手法の有効性が確認された。

6. 3次元配置領域の直方体への拡張

6.1 3次元配置問題 (直方体)

これまで3次元配置領域 R の形状を立方体に制限してきた。これ以降、 R の形状を x, y, z の大きさが u, v, w である直方体に拡張する。

[問題 3DR] 入力として、(1) モジュールの集合 $M = MC \cup MF$ (MC : 配置位置が固定され、直方体のマクロセルを形成するモジュールの集合, MF : 配置位置が固定されていない通常のモジュール), (2) ネットリスト N , (3) x, y, z の大きさが u, v, w である直方体の3次元配置領域 R が与えられる。このとき、目的関数として、配線長の総和 L を最小化する、各モジュールの、 R 内のスロット $sl_{0,0,0} \sim sl_{u-1,v-1,w-1}$ への重複のない配置を求めよ (モジュールの総数がスロットの総数より少ない場合、空きスロットがあってもよい)。 □

6.2 3次元交叉

3次元交叉に関しては、3次元配置領域が直方体になっても、提案手法をそのまま適用できる。その有効性に関しては、提案手法として3次元交叉と後で述べる直方体用の3次元突然変異を用いた手法と、従来手法として順序交叉と交換突然変異を用いた手法との比較で確認できれば十分であると考えられる。実験結果については6.4節で述べる。

6.3 3次元突然変異

3次元配置領域が直方体になると、分割した領域を回転する際、制約が生じる。そこで、回転を180度のみに制限する。

x, y, z の大きさが u, v, w である直方体の3次元配置領域において、領域内の xy 平面に平行な面のうち、3次元配置領域を $0 \leq z \leq i$ と $i+1 \leq z \leq w-1$ ($w-1$ は z の最大値) の2つの領域に分ける平面を、(直方体の3次元配置領域における) z 分割面と呼ぶ。 x, y 分割面も同様に定義する。

以下の [手続き 3DMT2] に、直方体の3次元配置領域に対する3次元突然変異の概要を示す。

[手続き 3DMT2]

- S1: 3次元配置領域内の x, y または z 分割面を1つ選び、3次元配置領域を2つの領域に分割する。
- S2: 選択された分割面がマクロセルを分割する場合、S1に戻って選択し直す。
- S3: 分割された片方の領域を、分割面に沿って180度回転をさせる。

3次元配置領域が立方体のときとアルゴリズムが変わっているため、提案手法の有効性を示すため、直方体用の3次元突然変異と、一般的な交換突然変異とを比較する。ここで交叉は両手法とも3次元交叉を用いる。

2×2×1の大きさのマクロセルを持つ、2×10×5から20×10×5までの10種類の直方体データ (R100, R200, ..., R1000) を、乱数を用いて作成した。用意した直方体データの各データ名、モジュールの配置サイズ、ネット数を表6に示す。R100~R800については、各ネットはランダムに2~5端子を接続し、R900, R1000については、各ネットはランダムに2~10端子を接続している。またすべてのモジュールはいずれかのネットに接続している。

この10種類のデータに対し、提案手法と比較手法をそれぞれ10回ずつ適用し、配線長の総和とCPU時間を比較した。配線長の総和とCPU時間を比較した結果を表7に示す。

実験の結果、総配線長で平均2.6%、提案手法の方がわずかに短く、CPU時間においては14.4%増と、従来手法と比

表6 直方体データ

Table 6 Data for rectangular solid region.

Data	Size (X×Y×Z)	Number of nets
R100	2×10×5	150
R200	4×5×10	250
R300	6×10×5	350
R400	5×8×10	500
R500	10×10×5	600
R600	3×10×20	700
R700	10×10×7	800
R800	8×20×5	900
R900	15×10×6	950
R1000	20×10×5	1000

表7 3次元突然変異2

Table 7 3-D mutation 2.

Data	Conventional method		Proposed method		Improved ratio [%]	
	L	Time	L	Time	L	Time
R100	924.4	11.0	970.3	11.9	-5.0	-8.2
R200	1702.2	14.5	1724.2	15.6	-1.3	-7.6
R300	2615.4	19.1	2444.8	22.4	6.5	-17.3
R400	4319.9	23.6	4064.6	27.9	5.9	-18.2
R500	5935.8	29.1	5441.7	34.0	8.3	-16.8
R600	9610.4	32.9	10466.5	34.7	-8.9	-5.5
R700	8508.4	44.0	7768.0	53.7	8.7	-22.0
R800	12213.6	51.2	11345.0	63.2	7.1	-23.4
R900	13138.9	54.8	12532.0	62.9	4.6	-14.8
R1000	15773.5	59.3	15730.3	65.1	0.3	-9.8

べて少し時間がかかっている。良い解を得るという観点で直方体の配置領域においても、3次元突然変異の有効性が確認された。

6.4 提案手法の有効性

提案手法として3次元交叉、3元突然変異 (直方体用) を、従来手法として1次元配列で遺伝子を表した順序交叉、交換突然変異を用い、これらの手法を、まず、小さいデータについて、最適解と比較する。

2×1×1, または2×2×1の大きさのマクロセルを持つ、5×1×2から3×3×2までの6種類の小さい直方体データ (R10, R12, ..., R18) を、乱数を用いて作成した。用意した小さい直方体データの各データ名、モジュールの配置サイズ、ネット数の関係を表8に示す。各ネットはランダムに2~5端子を接続している。またすべてのモジュールはいずれかのネットに接続している。

最適解については、マクロセル以外のモジュールの配置をすべての順序で置き換えるアルゴリズムを、64ビット版 CentOS 6.2 上で C++ (gcc version 4.4.6), および openmpi-gcc-1.4.3 を用いて実現した。実験は、Intel Xeon E5-4617@2.90 GHz (6コア) × 4 CPU, 128 GByte メモリの計算機上で、MPI を用いた 24 コアの並列計算で求めた。

提案手法と比較手法をそれぞれ10回ずつ適用し、配線長の総和とCPU時間を最適解と比較した。実験結果を表9に示す。最適解だけ計算機が違っているので、CPU時間は参考程度という意味で () を付けてある。小さいデータに対し、

表8 小さい直方体データ

Table 8 Data for small rectangular solid region.

Data	Size (X×Y×Z)	MC	Number of nets
R10	5×1×2	2	5
R12	3×2×2	2	8
R14	7×1×2	2	10
R16_2	2×4×2	2	10
R16_4	2×4×2	4	10
R18	3×3×2	4	10

表9 最適解との比較

Table 9 Comparison with optimum solution.

Data	Opt		Conventional method		Proposed method	
	L	(Time)	L	Time	L	Time
R10	10	(0.01)	10.1	3.9	10.1	8.0
R12	15	(0.12)	17.3	3.8	16.8	9.0
R14	17	(13.83)	20.1	3.8	19.8	8.9
R16_2	16	(3107.95)	20.0	3.9	16.7	9.1
R16_4	18	(15.49)	20.3	4.9	18.5	9.0
R18	25	(3273.02)	27.8	4.0	26.4	8.8

表 10 提案手法 (直方体の場合)

Table 10 Proposed method (In the case of rectangular solids).

Data	Conventional method		Proposed method		Improved ratio [%]	
	L	Time	L	Time	L	Time
R100	1162.3	13.6	970.3	11.9	16.5	12.5
R200	2192.3	40.1	1724.2	15.6	21.4	61.1
R300	3377.7	83.4	2444.8	22.4	27.6	73.1
R400	5421.5	137.5	4064.6	27.9	25.0	79.7
R500	7180.7	209.4	5441.7	34.0	24.2	83.8
R600	10876.6	300.4	10466.5	34.7	3.8	88.4
R700	10544.6	408.0	7748.0	53.7	26.3	86.8
R800	14309.0	528.9	11345.0	63.2	20.7	88.1
R900	15081.9	665.6	12532.0	62.9	16.9	90.5
R1000	17586.0	805.0	15730.3	65.1	10.6	91.7

配線長の総和の最適解と比較した場合、従来法では平均 11.9%増加したのに対し、提案手法では平均 6.3%の増加に抑えることができた。小さいデータに関しては、最適解に比較的近い解を求めていることが確認できた。

次に表 6 に示す、突然変異の実験で使った直方体の大きいデータに対し、提案手法として 3 次元交叉、3 次元突然変異 (直方体用) を、従来手法として 1 次元配列で遺伝子を表した順序交叉、交換突然変異を用い、これらの手法を比較する。

提案手法と比較手法をそれぞれ 10 回ずつ適用し、配線長の総和と CPU 時間を比較した。実験結果を表 10 に示す。

実験の結果、総配線長で平均 19.3%、提案手法の方が短く、CPU 時間においては 75.6%、提案手法の方が短い時間で求めることができおり、3 次元交叉、3 次元突然変異を用いた提案手法の有効性が確認された。

データが大きい場合、提案手法が従来手法に比べて非常に速くなっているのは、5.1 節で述べたとおり、全モジュール数に対して、交叉に関わる 3 次元選択領域に含まれるモジュールの数が、従来手法に比べて圧倒的に少なくなっており、重複遺伝子のチェックにかかる時間が短くなっているのが理由であると考えられる。提案手法では、交叉領域が小さくても、配線長に関して良い結果を得ることができている。

また、配置領域が立方体である場合 (従来手法に対して 39.6%減) に比べて、配置領域が直方体である場合 (従来手法に対して 75.6%減) は大幅に高速化しているように見えるが、立方体のデータに比べて直方体のデータが平均的に大きく、差が広がったためと考えられ、数値を直接比較することはできない。しかしどちらの手法も、従来手法に比べて十分高速であることが分かる。

7. おわりに

本論文では、マクロセルを考慮した 3 次元 FPGA 初期

配置問題に対し、遺伝子に 3 次元構造を持たせ、3 次元交叉、3 次元突然変異を導入した、3 次元 FPGA 初期配置のための遺伝的アルゴリズムを提案した。実験の結果、遺伝子を 1 次元配列で表した順序交叉と交換突然変異を用いた従来手法に対し、3 次元配置領域を立方体に制限した場合でも、直方体に拡張した場合でも、提案手法が配線長、および CPU 時間に関して、非常に有効であることが確認できた。ただし、提案する 3 次元交叉、3 次元突然変異によって理論的に最適解に到達可能かどうかは現時点で不明である。今後の課題としては、最適解への到達可能性を明らかにすること、アルゴリズムの並列化等によるさらなる高速化等がある。

参考文献

- [1] Pavlidis, V.F. and Friedman, E.G.: Three-Dimensional Integrated Circuit Design, Morgan Kaufman (2009).
- [2] Takahashi, S., Onodera, T., Hayashi, Y. and Kunio, T.: A New 3-D MCM Fabrication Technology for High-Speed Chip-to-Chip Communication: Vertically Connected Thin-film Chip (VCTC) Technology, *Proc. 1995 Symposium on VLSI Technology*, pp.135–136 (1995).
- [3] Sherwani, N., Bhingarde, S. and Panyam, A.: Routing in the Third Dimension — From VLSI to MCMs, IEEE Press (1995).
- [4] Sarrafzandeh, M., Wong, M. and Yang, X.: Modern Placement Techniques, Kluwer Academic Publishers (2002).
- [5] Cong, J. and Luo, G.: A Multilevel Analytical Placement for 3D ICs, *Proc. Asia and South Pacific Design Automation Conference*, pp.361–366 (2009).
- [6] Hsu, M.-K., Chang, Y.-W. and Balabanov, V.: TSV-Aware Analytical Placement for 3D IC Designs, *Proc. Design Automation Conference 2011*, pp.664–669 (2011).
- [7] Kahng, A.B., Lienig, J., Markov, I.L. and Hu, J.: VLSI Physical Design: From Graph Partitioning to Timing Closure, Springer (2011).
- [8] Breuer, M.A.: A Class of Min-Cut Placement Algorithms, *Proc. Design Automation Conference*, pp.284–290 (1977).
- [9] Suaris, P.R. and Kedem, G.: A Quadrisection-Based Combined Place and Route Scheme for Standard Cells, *IEEE Trans. Computer-Aided Design*, Vol.8, No.3, pp.234–244 (1989).
- [10] Hentschke, R., Flach, G., Pinto, F. and Reis, R.: Quadratic Placement for 3D Circuits Using Z-Cell Shifting, 3D Iterative Refinement and Simulated Annealing, *Proc. SBCCI '06*, pp.220–225 (2006).
- [11] Obenaus, S.T. and Szymanski, T.H.: Gravity: Fast Placement for 3-D VLSI, *ACM Trans. Design Automation of Electronic Systems*, Vol.8, No.3, pp.298–315 (2003).
- [12] Sui, W., Dong, S. and Bian, J.: Wirelength-driven Force-directed 3D FPGA Placement, *Proc. GLSVLSI '10*, pp.435–440 (2010).
- [13] Wakabayashi, S.: A Genetic Algorithm for Rectilinear Steiner Tree Problem in VLSI Interconnect Layout, *IPSI Journal*, Vol.43, No.5, pp.1315–1322 (2002).
- [14] 高津尊是, 若林真一, 永山 忍: LSI フロアプランニングに対する遺伝的アルゴリズムとタブー探索に基づく並列

アルゴリズムとその PC クラスタ上での実現, 情報処理学会研究報告 SLDM-128, pp.31–36 (2007).

- [15] Nunna, K.C., Mehdipour, F. and Murakami, K.: Early Stage Power Management for 3D FPGAs Considering Hierarchical Routing Resources, *Proc. GLSVLSI '13*, pp.31–36 (2013).
- [16] 寺井秀一: VLSI デザインオートメーション入門, コロナ社 (1999).
- [17] 電気学会編: 遺伝的アルゴリズムとニューラルネットワークスケジューリングと組合せ最適化, コロナ社 (1998).



大村 道郎 (正会員)

1985 年広島大学工学部第二類 (電気系) 卒業. 1991 年同大学大学院工学研究科博士課程後期単位修得後退学. 同年広島工業大学工学部助手, 講師を経て, 2000 年同大学准教授. 工学博士. VLSI 設計自動化, VLSI 設計教育

に関する研究に従事. IEEE, ACM 各会員.



太田 雅也

2011 年広島工業大学工学部電気・デジタルシステム工学科卒業. 2013 年同大学大学院工学系研究科博士前期課程修了. 現在, 西日本旅客鉄道勤務. 在学中は VLSI 設計自動化に関する研究に従事.