親の形質を詳細に利用する遺伝的アルゴリズムを用いた電子系DAに於ける部品配置手法

多胡 誠久、白石 洋一
群馬大学 工学部 情報工学科

電子系レイアウトDA（Design Automation）に於ける部品配置問題は、組合せ最適化問題として定式化できる。しかし、大規模配置問題に対しては、実用時間内にその最適解を得ることは不可能である。従って、従来より様々な発見的手法が用いられてきた。ここでは、局所最適化に陥らずにいかにして大域的最適解に到達するかが問題となる。最近、確率を用いて局所最適解から抜け出すSA（Simulated Annealing）とGA（Genetic Algorithm）が重要になりつつある。配置処理にSAを適用した例は一般的になりつつあるが、GAを適用した例は未だ少なく大きな効果を上げているとは言えない。 本稿では、従来注目していなかった「形質遺伝」という性質に深く注目して染色体を遺伝させるGAを開発し、配置処理に用いて実験、評価した結果について論ずる。

A Placement method based on the Genetic Algorithm which effectively utilizes the Characteristics of Parents in the Electronic Design Automation

Masahiasa Tago, Yoichi Shiraishi
Department of Computer Science, Gunma University
E-mail:{tago,siraisi}@keim.cs.gunma-u.ac.jp

Placement problems of components in the electronic design automation are generally formalized as combinatorial optimization problems. However, if the problem size is very large, it is impossible to obtain the optimum solution within a practical processing time. Therefore, various kinds of heuristics have been devised and implemented. Here, the serious and difficult problem is how to obtain a globally optimum solution not trapped in a locally optimum solution. These days, stochastic algorithms such as SA (Simulated Annealing) and GA (Genetic Algorithm) are devised and utilized in order to get out from a locally optimum solution. The placement method based on the SA is generally regarded as useful and actually implemented in various layout DA systems, however, not so many placement methods based on the GA are suggested and applied to practical problems. In this paper, a new type of genetic algorithm which inherits the genes to their descendants concentrating on the "characteristic inheritance", which was not noticed before, is suggested and the experimental results are shown.

1 配置問題

1.1 VLSIの設計

VLSIの設計は、論理設計、電気的設計と物理的設計に分けられる。物理的設計では論理設計結果の素子の結線関係と、電気的設計結果の部品情報とから実際に部品をチップ上にレイアウトする。物理的設計は配置処理、配線処理の順に行なわれる。

1.2 配置処理の概要

配置領域に配置する対象は、論理回路を実現する素子である。配置処理時には、複数の論理ゲートをまとめたセルを最小単位として扱う。VLSIの設計手法は複数存在し、重要な手法の一つに階層設計手法がある。この手法は、配置対象をセル、複数のセルで構成するブロック、複数のブロックで構成するチップの3階層に分けて、再帰的に設計する。扱う最大規模はブロック数で数百、各ブロックを構成するセル数で数万程度である。図1は階層設計によって構成されたVLSIの例である。本稿の配置処理は階層設計手法を対象としている。以下ではセル及びブロックをまとめて部品と呼ぶ。

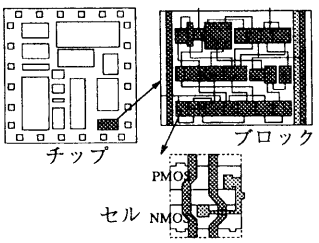

図1:VLSIチップの階層設計方式

1.2.1 配置処理の目的関数

VLSI設計の目標は面積最小化である。配線処理では配線経路を実現する領域を必要とする。配線長合計の短縮は配線領域削減につながり、最終的に面積縮小になる。従って、配置処理の目的関数は配線長合計最小化である。

配置処理では、配線長合計は仮想的に決めた配線長で近似する[2]。配線の要求は、ネットと呼ぶ同電位にすべき端子の集合である。以下では端子は部品の中心にあると見做す。

1.2.2 部品の配置モデル

VLSI設計では配置すべき部品数が非常に大きく、また、配線領域を考慮しなければならない事から、単に隙間なく部品を配置することには意味がない。そこで通常は配置領域に複数の部品列を設け、各部品列間で部品が重ならないように、また、各々の部品列内で隣接する部品が重ならないように配置モデルをとる。こうすることで、部品の形状やサイズに左右されない配置が可能になる。例を図2に示す。この図では3部品列に部品を配置している。

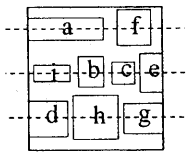

図2: 配置モデル

2 遺伝的アルゴリズムと配置問題

2.1 遺伝的アルゴリズム

遺伝的アルゴリズム(以下GAと呼ぶ)とは、生物進化(選択淘汰、突然変異)の原理に着想を得た最適化アルゴリズムであり、多点情報を利用した確率的探索法の一種である[1]

GAでは、問題の解を、解の質を左右する形質(特徴)の集合であると考え、形質を遺伝子として表現し、解は遺伝子の集合で構成した染色体で表現する。染色体は個体とも呼び、個体の集合を個体集合、集合の個体数を人口と呼ぶ。ある時期の個体集合を世代と呼び、ある世代に遺伝的操作を行なって新しい個体集合を作ることを世代交代と呼ぶ。

GAでは、優良な遺伝子を後世代に残す遺伝的操作を用いて世代交代を重ねることにより、優良な遺伝子の集約が図れるという発想を基にしている。そこで、問題の複数の解をそれぞれ1染色体として表現し、その個体集合に対して世代交代を複数回行なった後に、優良な遺伝子を多く持つ最も優良な個体を最良解として選択する。

遺伝的操作は交叉、突然変異、および選択という3種類からなる。交叉は子供を作る操作で、親の優良な形質を子供に遺伝させることができれば、解の探索効率は飛躍的に向上する。突然変異は、個体集合が持たない新しい遺伝子の導入と、特定の遺伝子が集合内に広まることによる個体集合の多様性の欠如を回避する働きをする。

2.2 遺伝アルゴリズムの処理

初期世代を第0世代とし、この個体集合をinitial setと呼ぶ。第n世代の個体集合をG_nで表し、人口を定数Pで表す。交叉のために個体集合から2つの個体を選択する関数を交叉選択と呼びδで表す。交叉を関数ϕで表し、また突然変異を関数μで表わす。次の世代を担う子供を作るために選択するペアの数は$P \times R_\phi$で表し、定数R_ϕ $(0 \leq R_\phi \leq 1)$を交叉率と呼ぶ。交叉や突然変異の結果、解としてふさわしくない遺伝子を持った個体が生まれる可能性があり、そのような個体を致死個体と呼び以下deadと記す。個体は評価関数σで評価する。評価関数は交叉選択時などに用いる。2つの個体を交叉させてできた個体を子供(offspring)と呼び、第n世代に遺伝的操作を行なって得られた全ての子供に一定の突然

変異確率 R_μ $(0 \leq R_\mu \leq 1)$ で突然変異を適用し、その結果得られる個体集合を G'_n と表す。G_n と G'_n から次世代 G_{n+1} を選択することを淘汰選択と呼び、関数 τ で表す。また、ξ を終了条件を定義する関数とする。一般に終了条件には世代交代数などが当てられる。以下に GA のアルゴリズムを記す。

GA のアルゴリズム

$i \leftarrow 0$
$G_i \leftarrow \{initial\ \ set\}$
$do\{$
　　$X \leftarrow Y \leftarrow G'_i \leftarrow \{\}$
　　$do\{$
　　　　$\{x, y | x, y \in G_i\} \leftarrow \delta(G_i, X, Y, \sigma)$
　　　　$X \leftarrow \{X, x\}$
　　　　$Y \leftarrow \{Y, y\}$
　　　　$child \leftarrow \phi(x, y)$
　　　　$if(random < R_\mu)\ child \leftarrow \mu(child)$
　　　　$if(child \neq dead)\ G'_i \leftarrow \{G'_i, child\}$
　　$\}while(\ \ |G'_i| < P \times R_\phi\ \ )$
　　$G_i \leftarrow \tau(G_i, G'_i)$
　　$i \leftarrow i + 1$
$\}while(\ \ \neg\xi(i, G_i)\ \ )$

2.3 部品配置のコード化

2.3.1 1 次元配列化

遺伝的アルゴリズムを使って配置をする場合には、一般的に、2 次元の問題を 1 次元の配列にコード化する。コード化には配置領域の部品列を用い、各行を図 3や図 4のようにつなぎあわせて 1 次元の配列に直す。図 3を simple zigzag 方式 [5]、図 4を raster 方式と呼ぶ。

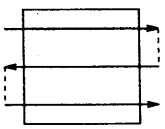

図 3:simple zigzag 方式

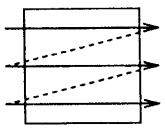

図 4:raster 方式

2.3.2 染色体表現

部品列に於いて、各部品を配置する位置をスロットと呼ぶ。本稿では、スロットには simple zigzag 方式の順に、0 からの昇順の自然数を割り当てる。図 5は図 2の配置を 1 次元配列に直してスロットと部品の名前を対応させたもので、染色体を表現する。

スロット番号	0	1	2	3	4	5	6	7	8
部品名	a	f	e	c	b	i	d	h	g

図 5: 染色体

2.4 従来の交叉操作

GA の配置処理への導入の歴史は浅いので、まだ一般的な GA を配置処理問題用にチューニングしている段階である。一般的な GA の操作を配置問題に取り入れた場合に最も問題となるのは、交叉操作によって部品の未配置や 2 重配置が生じることである。この場合には個体が致死個体となり、解として不適格になる。

order 交叉、PMX 交叉、cycle 交叉 [4]、などの交叉手法が配置処理に用いられている。いずれも 1 部品を遺伝子として表現し、部品を過不足なく子供に遺伝することをメインにした交叉手法であり、配置処理問題特有の形質を熟慮した遺伝子の捉え方をしていない。

このように、1 部品を遺伝子と捉えても、評価値の良い個体を交叉時に選択することにより、優良な形質の一部は遺伝されるため、収束速度を問題にしなければ、ある程度の結果を期待できる。しかし、配置問題特有の形質を十分考慮した遺伝子の捉え方をし、交叉操作に反映させることができれば、解の質向上と収束速度の向上が期待できる。

3 提案する交叉操作

3.1 基本的考え方

配置処理の目的関数である配線長は、結線される部品の位置関係に大きく依存する。従って、位置関係が意味を持つ部品の集合 (以降クラスタと呼ぶ) を遺伝子として捉え、クラスタを直接子供に遺伝できれば、解の収束速度と質向上が期待できる。また、1 部品を遺伝子としていた従来法では、ある遺伝子が解に対してどれほど重要性を持つかを判定することは不可能である。しかし、例えば遺伝子をネットと見做せば、ネットは配線長という目的関数値を用いて、個体を形成するネットどうしの比較や、両親の同一ネットの比較が可能であることから、ある遺伝子の解に対する相対的な重要性を判定することが可能となる。この性質は、良い形質を遺伝させる交叉操作の構成に有力である。

3.1.1 クラスタ

部品のクラスタ化にはネットや部品間の配線本数を考慮する接続度 [6] などを用いることが考えられる。本稿ではネットを基にしたクラスタ化を考える。全ての部品の配線要求 (全てのネット) は配置フェイズの入力として与えられ、また、部品配置では不変である。

3.1.2 ネットによるクラスタ化の利点と欠点

ネットによるクラスタ化の利点は、配置目標を仮想配線長合計最小化とする場合、個体の目的関数値は全ネットの目的関数値の合計であるので、各ネットと個体の目的関数値が同時に得られることである。また、信号遅延などの特性はネット単位で与えられるので、目的関数に

仮想配線長合計の他に、電気的特性などを含めることも可能と考えられる。

欠点は、1つの部品が複数のネットに含まれる可能性が高く、全てのクラスタが完全な形で遺伝されるわけではないということである。しかしそれは逆に、遺伝時に両親の複数ある遺伝子の中の一部の遺伝子を破壊することで、突然変異と同様の効果をもたらすことが期待できる。これは、単なる両親の遺伝子の合成によって得られる探索範囲よりも、広い探索を行うことに相当する。

3.1.3 2重配置問題

クラスタが図6のような関係になっている場合に、clusterA、clusterBを独立な遺伝子と考えて遺伝させると、斜線部の部品を子供に2重配置する事になり、致死個体を発生させる。これを避けるために本手法では各クラスタに遺伝の優先順位をつけ、クラスタ間に共通な部品がある場合には、共通部の部品は優先順位の高いクラスタによって遺伝させ、優先順位の低いクラスタの遺伝時には、その部品を配置済みとする。

遺伝にクラスタ内部品相対位置を使用すれば、優先順位に関係なく複数のクラスタ間に共通の部品が活かせるが、図7の様に、共通部品を2つ以上の異なるクラスタと持ち合うクラスタ(clusterC)を遺伝させる際に、持ち合う複数のクラスタを加味して配置場所を探すことは困難である。従って、部品を遺伝をさせる際には、各部品が親のスロット番号をそのまま受け継ぐ形を取る。既に優先するクラスタの遺伝の結果、そのようなスロットに遺伝させることができない場合には、最寄りの空きスロットに置く。

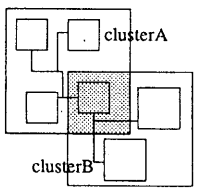

図6: クラスタの重なり1

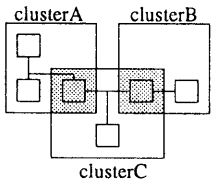

図7: クラスタの重なり2

3.2 アルゴリズム

3.2.1 対象

外部端子(位置可変)、外部端子(位置固定)、矩形の配置領域に全て一定の高さを持つ矩形部品を配置する対象とする。

3.2.2 目的関数

部品配置の評価は、クラスタ毎に疑似スタイナー木を使用して仮想配線長を求め、仮想配線長合計とする。

3.2.3 交叉 (ϕ)

交叉は3つのフェイズに分ける。第1のフェイズは各クラスタについて、どちらの親から遺伝させるかを決定し、第2のフェイズは、それらのクラスタの遺伝順序を決定する。そして第3のフェイズでは、遺伝時に起こり得る部品の2重配置を避けながら遺伝を実際に処理する。

3.2.4 突然変異 (μ)

突然変異度定数 D_μ $(0 \leq D_\mu \leq 1)$ を用い、スロットの数 $\times D_\mu$ 個の無作為に選択した部品のペア交換によって行なう。

3.2.5 交叉選択 (δ)

仮想配線長で評価するため、目的関数値の小さいもの程優良な個体とする。よって交叉において個体の選択確率は目的関数値の逆数に応じて決定する。[3]。

3.2.6 淘汰選択 (τ)

第 $i+1$ 世代 G_{i+1} の構築のための淘汰選択は、第 i 世代の子供の集合 G_i' と、G_i から目的関数値の逆数に対応した選択確率で $P(=$ 人口$) \times (1 - R_c(=$ 交叉率$))$ 個を選択したものを合わせる。

3.3 具体例

simple zigzag 法を用いたコードを使用し、ネット毎の遺伝を行なう交叉(net 交叉)の具体例を以下に記す。

9つの部品 {a, b, c, d, e, f, g, h, i} からなる配置問題があり、その問題には net1{a, b, c}、net2{d, e, f}、net3{c, g, h}、net4{a, i} の4つの配線要求(ネット)があるとする。既に個体集合が存し、交叉選択を行なったところ、図8の father と mother が選択されたとする。この father、mother をコードで表したものが、図9の fahter、mother である。図9でスロットを結んでいる線分はネットを表している。

交叉の第1フェイズとして father、mother の各ネットを比較評価した結果、net1 および net3 が father から、net2 および net4 が mother から遺伝することが決まったと仮定する。

次に交叉の第2フェイズとしてネットの評価値を考慮した結果、net1、net2、net3、net4 の順で子供に遺伝させることが決まったと仮定する。

以上の仮定の下で、交叉の第3フェイズとして net1、net2、net3、net4 の順にネット毎に部品を子供のスロットに割り付ける手順について図10～13に図示する。図中で実線で結ばれたスロットはその図の遺伝で親から遺伝する遺伝子を表し、破線で結ばれたスロットは既に他の優先順位を持つネットによって配置されている部品を指す。また、網がけした部品は、本来遺伝すべきスロット

に既に他の部品が入っていたために、最寄りの空きスロットに配置した部品を指す。

以上の交叉操作の結果得られる配置を図14に図示する。

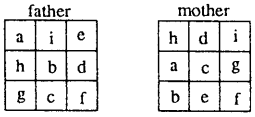

図 8: 交叉選択された両親

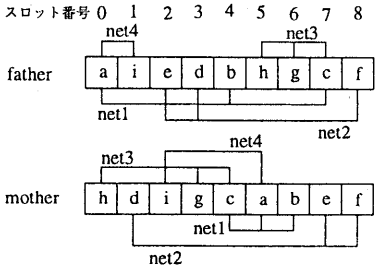

図 9: 両親のコード

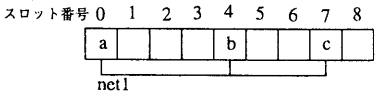

図 10: net1 の遺伝

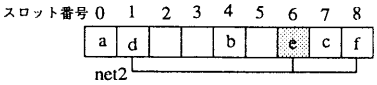

図 11: net2 の遺伝

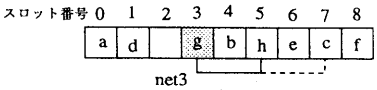

図 12: net3 の遺伝

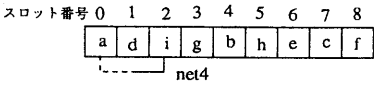

図 13: net4 の遺伝

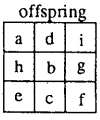

図 14: 交叉結果としてできた子の配置

4 実験

実験では、MCNC のベンチマークの fract.yal、primary1.yal、primary2.yal、sclib.yal、db.yal を一部変更して使用した。配線長最小化に評価の重点を置いたので、部品面積が領域面積の 70 ～ 75% になるような配置領域を初めに決め、部品の平均のサイズをもち、意味を持たないダミー部品を配線領域確保の代わりとして、配置領域の 20 ～ 25% ほど混入して実験を行なった。ただし、配線領域としての部品列の間隔は取らなかった。各データの外部端子を含めた部品数は fract で 188 個、primary1 で 1163 個、primary2 で 3920 個である。また、コード化は simple zigzag 式を採用した。

GA の解探索終了条件は世代交代数とし、同じ染色体を持つ異なる個体の存在を、あらゆる場合において許した。パラメータは、交叉率 (R_ϕ) を {0.7, 0.9}、突然変異確率 (R_μ) を {0.0, 0.01, 0.1}、突然変異度 (D_μ) を {0.01, 0.1}、人口 (P) については、データ fract には { 100, 1000 }、primary1 には 500、primary2 については 100 を用いた。

上に挙げた様々のパラメタと各関数を組合せて、のべ約 1660 時間計算機を使用した。同一パラメタによる実験結果は fract で 4 回、primary1、primary2 で 2 回づつ行なった。計算機は SUN の SPARC station5,10,20,LX、日本電算の JS5/85,JS5/110 を使用し、プログラム記述は C++ で行なった。

本手法の実験比較対象としては、simple zigzag 式のコードを用いた cycle crossover[4] を用い、実験は双方の手法ともにできるだけソースコードを共有させたプログラムを用いた。

表 1はデータ fract(D_1),primary1(D_2), primary2(D_3) に対する処理結果である。ϕ は交叉法を表し、世代数は世代交代数を表す。処理は世代数に標記した世代交代数にかかった処理時間を表す。異種計算機を併用して実験しため、処理時間は、JS5/85 での処理時間に変換した値である。最良値の単位は、配置処理で部品を配置する際に用いる DA 格子である。また最良値の右に記した値は、最良値を得た世代である。3 つのデータを用い、cycle 交叉と本手法の差が明らかであると考えるのに十分なだけの世代交代を行なったが、fract(D_1) については cycle 交叉の解がほぼ収束するまで世代交代を行なった。

図 15～ 17は、様々なパラメタを試したうちで、最も良い値を出した組合せの結果であり、世代交代過程に於ける、最良個体の目的関数値のプロット結果である。パラメタの詳細は図 15～ 17中にそれぞれ記した。

図 15と表 1からは、問題規模が小さい場合には cycle 交叉と net 交叉は同等程度の解に収束し、収束速度は net 交叉が優れていることが確認できる。図 16,17と表 1からは、問題規模が大きい場合には cycle 交叉は形質の遺伝がうまく行なわれていないせいか、fract1 の様な解曲線を描かず、収束速度が遅いことが確認できる。それに対して net 交叉は、問題の大きさにあまり左右さ

れずに初期収束が速いことが確認できる。また、primaty1 と primary2 についての cycle 交叉の収束速度は net 交叉に比べて異常に遅いので、実用時間内に解を得るということを考慮すると、net 交叉の方が良解を得ていると言える。

以上のことから、net 交叉は従来の交叉と比較して、収束速度が速く、解の質も同等以上のものを得られることが期待される。

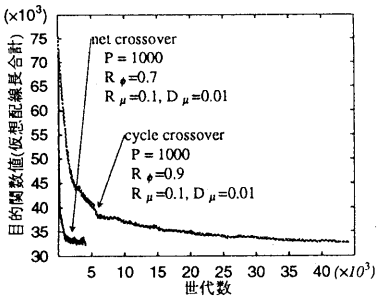

図 15: D_1:fract 使用の結果

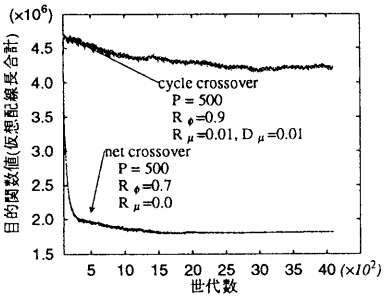

図 16: D_2:primary1 使用の結果

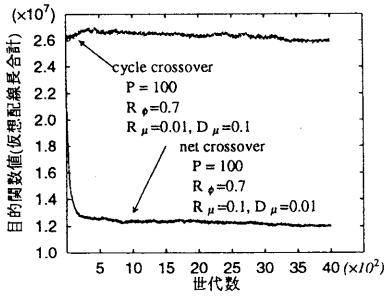

図 17: D_3:primary2 使用の結果

表 1: 実験結果

	ϕ	世代数	処理 (秒)	最良値 (世代)
D_1	net	4000	17000	32127 (3971)
	cycle	44100	165000	32451 (42126)
D_2	net	4000	57000	1791645 (1632)
	cycle	4000	50000	4141320 (2815)
D_3	net	4000	41000	11927142 (3784)
	cycle	4000	28000	25740224 (3643)

5 まとめ

- GA の配置問題に関する形質をクラスタという部品集合の単位で捉え、交叉によるクラスタ毎の遺伝を提案した。
- ネットを考慮した部品集合をクラスタとし、ネット毎の遺伝を行なう net 交叉を提案し、評価した。

6 今後の課題

- ネットを考慮した部品集合をクラスタとし、目的関数に電気的特性などの仮想配線長以外の制約を盛り込んだ場合の評価。
- 接続度など、ネット以外でクラスタ化をした場合の交叉方法の考案と評価。

参考文献

[1] 北野宏明編, 「遺伝的アルゴリズム」, 産業図書, 1993.

[2] 高井英造, 鈴木誠道編, 「講座・数理計画法 11/ 数理計画法の応用 (実際編)」, 産業図書,

[3] JAMES P. COHOON and WILLIAM D. PARIS, "Genetic Placement", *IEEE Trans. computer-Aided Design*, vol. CAD-6, NO.6. pp. 956-964, November 1987.

[4] K. Shahookar and P. Mazumder, "GASP - A GENETIC ALGORITHM FOR STANDARD CELL PLACEMENT", *IEEE Trans.*, pp. 660-664, 1990.

[5] T. Kozawa, H. Terai, T. Ishii, M. Hayase, C. Miura, Y. Yamada, and Y. Ohno, "Automatic Palcement Alghrithms for High Packing density VLSI", *IEEE Trans.*, pp. 175-181, 1983.

[6] Donald M. Schuler and Ernst G.Ulrich, "CLUSTERING AND LINEAR PLACEMENT", *Proceedings of the International Conference on Circuits and Computers* , pp.50-56,June 1972.