ハイブリッド GA による LSI セルの Congestion-Driven 並列配置手法とその評価*

吉川 雅弥[†]・寺井 秀一[‡]・山内 寛紀[‡]

Congestion-Driven Parallel Placement Procedure for LSI Cells Based on Hybrid Genetic Algorithm*

Masaya YOSHIKAWA[†], Hidekazu TERA[†] and Hironori YAMAUCHI[‡]

A congestion-driven placement technique for LSI cells through hybrid genetic algorithm is presented. In particular, the procedure consists of a two-level hierarchical placement procedure and a hybrid genetic algorithm, in which the algorithm is combined with the approach of searching locally for an optimal solution. For selection control, new objective function are introduced to each phase for dispersing congestion, and a parallel processing technique suited to hierarchical placement is proposed as an effective approach to accelerating the processing speed of genetic algorithm-based placement. Regarding to total virtual wire length and wire congestion, the ratio of the proposed approach to the conventional one is 0.76 and 0.9, respectively. As a result, the effectiveness of the suggested approach is shown through a comparison with the conventional one.

1. はじめに

組合せ最適化問題は、その多くが NP-困難であることが知られており、多項式時間アルゴリズムが存在しないと予想されている。そのため、大規模な問題例に対しては、現実的な時間で厳密な最適解を求めることがきわめて困難である。しかし、その応用範囲は多岐にわたっており、理論的に最適解を多項式時間で求める解法が存在しなくても、経験的に比較的良好な解を求めるアルゴリズムは数多く提案されている。

このような準最適解を効率よく求めることができるアルゴリズムにシミュレーテッドアニーリング (Simulated Annealing: SA) [1] や遺伝的アルゴリズム (Genetic Algorithm: GA) [2-19] がある。

SA は統計力学において、溶解状態にある物質を冷却

して結晶状態に到達させるプロセスからヒントを得たアルゴリズムであり、Fig. 1 に示すように局所解へ陥るのを防ぐために、山登り法に確率的な遷移を導入している。

一方 GA は、生物の進化機構にアイデアを得て Holland[2] により発案されたアルゴリズムであり、Fig. 2 に示すようにあらかじめ決められた数の個体の集合を常に保持し、選択・交叉・突然変異という三つの遺伝オペレー

Fig. 1 Point-based search

Fig. 2 Population-based search

* 原稿受付 2000 年 9 月 26 日

[†] 立命館大学 総合理工学研究機構 VLSI センター
VLSI Center, Research Organization of Science and Engineering, Ritsumeikan University; 1-1-1, Nojihigashi, Kusatsu city, Shiga 525-8577, JAPAN

[‡] 立命館大学 理工学部 電気電子工学科 Department of Electrical and Electronic Engineering, Faculty of Science and Engineering, Ritsumeikan University; 1-1-1, Nojihigashi, Kusatsu city, Shiga 525-8577, JAPAN

Key Words: hybrid genetic algorithm, parallel processing, congestion-driven placement, hierarchical placement, LSI layout design.

ションを適用することにより世代交代を繰り返し、その集団を進化させ解を導く。

ここでSAに代表される局所探索法(1点探索アルゴリズム)は、局所的な探索能力は優れている反面、大域的な探索能力が弱いという欠点がある。一方、多点探索アルゴリズムであるGAは交叉演算を中心に新しい探索点を生成するため、良い解の近傍における系統的な探索は苦手である。このような相補的な関係にあるアルゴリズムを組み合わせたハイブリッド探索は、さらに解の探索能力を向上させることが期待できる。そのため、良い解の近傍における系統的な探索を局所探索に行わせ、大域的な探索をGAが行うというハイブリッド化が今までいくつか提案されてきた[11-16]。

本論文ではこのハイブリッド型GAをベースとする新しいLSIセルのCongestion-Driven並列配置手法を提案する。セルとはAND/ORなどの基本ゲートやフリップフロップなどの比較的小規模の論理エレメントのことである。LSI (Large Scale Integrated circuit: 大規模集積回路)の設計は大きく、論理設計(logic design)とレイアウト設計(layout design; physical designともいう)の二つの工程にわかれ、本論文で展開するセル配置はレイアウト設計工程に属する。

レイアウト設計において面積の最小化は、LSIの性能やコスト、製造歩留まり(良品率)をよくすることにつながるため重要な指標となる。チップの面積は個々のセルの面積の総和と配線に必要なスペースを合わせたものである。セルの面積は論理設計の結果によって使用するセルの種類が決められているため変えることができず、配線のスペースを小さくすることが配置設計の課題となる。ここで、配線スペースと配線の長さには相関があることが知られており、配置の良さを評価する基準として、一般に配線の長さの総和を用いる。

しかし、配置処理を行っている段階においては、個々のセルを接続する配線経路は決まっていないため、正確な配線の長さを知ることはできない。そこで、従来から配線長の推定値(以後、仮想配線長と呼ぶ)としてFig. 3に示すようなネット(同じ信号で接続される端子の集合)の最小外接矩形の半周長が用いられている。この推定値を用いるとセル配置問題は、“すべてのネットの仮想配線長の総和が最小になるセルの配置位置を決定する問題”となり、この条件を満たす配置が最適配置ということに

Fig. 3 Virtual wire length

なる。

また、Congestion-Driven配置とは、配線の極端な集中を分散化(チップ全体に散りばめる)させることによって、配線が一部分に集中することを防ぎ、routability(配線可能性)の高い配置を実現する手法のことである。さらに、LSIの微細化・高集積化が進むと、各設計工程内のイタレーションと工程間フィードバックループの回数が急増するため処理のQTAT化(Quick Turn Around Time)も必須課題の一つである。

GAをLSIレイアウト問題に適用した報告として、回路分割[3]やフロアプラン[4,5]に関する報告がある。またセル配置問題に適用したものとしてスタンダードセルを対象にGAパラメータの最適化問題を扱ったもの[6]や、アナログLSIを対象に、回路図上の素子の相対的な位置関係と素子の隣接性の制約下で仮想配線長と面積を最小化する問題を扱ったもの[7]がある。なお、商用のLSI設計システムに組み込まれているセル配置手法は、基本的に局所探索法のみをベースとしており、GAを用いたものは未だ実現されていない。

以下に論理LSIを対象としたセル最適配置決定アルゴリズムを下記の点に絞って述べる。まず、(1)配置手法の基本アルゴリズムとして概略配置法と詳細配置法の二つからなる階層化配置手法を導入し、各配置手法において「GA」と「局所探索法」を組み合わせたハイブリッド化を施すことによって、セルの最適な配置位置を決定する手順を示す。つぎに(2)遺伝オペレーションでの重要な項目の一つである選択(淘汰)制御に関して、概略配置では配線混雑の分散化を、詳細配置では局所的な配線混雑緩和を目的にした新しい評価関数を導入する。さらに(3)概略配置と詳細配置のそれぞれに適合した並列処理方式を考察し、これらを並列計算機上に実装し、高速な配置処理システムを実現する。

本論文で展開する階層化配置法のもとで、GAと局所探索法を組み合わせ並列計算機へ実装し、その効果を実証した報告はこれまでなされていない。

2. ハイブリッドGAをベースとする階層化配置手法

2.1 配置手法の概要

本論文で提案する配置手法は、概略配置(outline placement)と詳細配置(detail placement)の2ステップで構成する。まず概略配置では、Fig. 4(1)に示すように、配置領域を四つに分割し、各分割領域に対しセルの割り振りを行い、これを繰り返すことによって配置を段階的に決定していく。すなわち、セルを四つのグループに分け、各グループをLSIチップの四つの領域にそれぞれ対応させる。各部分領域とそのセルグループに対して上と同様の4分割処理を逐次繰り返し、部分領域とセルグループを細かくしていく。

Fig. 4 Hierarchical placement

詳細配置では、Fig. 4(2)に示すように、LSI上に2次元グリッド領域を設定し、概略配置の最終段階で得られている各部分領域内のセルの最適な配置位置を決定する。また各配置手法において基本アルゴリズムにGAを適用する。

ここで、ハイブリッド探索をレイアウト問題に適用した例としては、アナログLSIを対象に、初期段階をGAで探索し、途中から局所探索法の一つであるTabu Search[20]に切り替える方法が報告されている[7]。これは、処理の前段にGA、後段に局所探索法を用いたもので、進化ループと局所探索は独立している。本論文では、局所探索を進化ループの中に組み込んだハイブリッド探索、すなわち、個体を評価する際に個体が表す解をそのまま評価するのではなく、その個体を初期値として局所探索を行い、局所探索によって得られた解をGAにおける各個体の評価とするハイブリッド探索[16]をLSIセル配置問題に応用した。具体的な処理フローをFig. 5に示す。

Fig. 5 Hybrid GA

2.2 概略配置におけるコーディング

概略配置では、遺伝子座にCell ID（個々のセルを区別するための番号）を、遺伝子にPlacement Area（配置領域名）を対応させる。すなわち、Cell ID順に分割

領域名を保存する形で個体を表現する。Fig. 6は概略配置の途中ステップで、8個のセルをAからDの四つの領域に分割した例を示しており、'2'と'8'のセルがArea-Aへ、'5'と'6'がArea-Bへと分割されている。

Fig. 6 Example of coding in outline placement

2.3 概略配置における遺伝オペレーション

2.3.1 選択

GAにおいて次世代での集団形成メンバーとなる個体を継続させること、すなわち個体の選択を制御する評価関数の設定は重要である。本手法では、routabilityの高い配置の実現を目標に、以下に二つの評価を組み込んだ新しい評価関数を考案した。

(a) ネットの仮想配線長の総和

ここでは、現在処理を行っている領域外のセルを含んだネットの最小外接矩形長の1/2の総和を用いる。具体的には、Fig. 7に示すように現在処理を行っている領域（横の長さをa、縦の長さをbとした場合）内にあるCell-1・Cell-2と領域外にあるCell-3とを接続しているNet-tの仮想配線長は、(1)式のように求めることができる。

$$\text{length}(t) = \frac{1}{4}a + \frac{1}{2}a + \frac{3}{4}b + \frac{1}{2}b = \frac{3}{4}a + \frac{5}{4}b \quad (1)$$

Fig. 7 Example of estimating total virtual wire length of a net

(b) カットライン偏差

分割を繰り返していく過程での各領域（Fig. 8では、Area-A,B,C,D）をまたがるカットライン（領域間配線）の中で最大のカットライン数から最小のカットライン数を引いた値（以後、カットライン偏差と呼ぶ）を用いる。

Fig. 8は、ある分割におけるカットライン偏差を示したもので、(1)のカットライン偏差は3であり、(2)の

Fig. 8 Deviation of across-border cutline

カットライン偏差は0である。このカットライン偏差が小さいほど配線の集中を抑えることができ望ましい分割であるといえる。

このように、概略配置で用いる評価関数は配線混雑の平準化を目的としたもので、以下では、“MWC (Measure for Wide area wire Congestion)”と呼ぶことにする。MWCは、各ネットの最小外接矩形長の1/2と各領域をまたがるカットライン偏差の和をとった評価関数で、(2)式を評価式として用いた。

$$f(t) = \left\{ \alpha \times \sum_{i=0}^{allNets} netLength(i) \right\} + (maxBorder - minBorder) \quad (2)$$

ただし、 $netLength(i)$ は*i*番目のネットの外接矩形長の1/2とし、各領域をまたがるカットライン数の最大値、最小値をそれぞれ $maxBorder$ 、 $minBorder$ とする。また、*t*を個体、 α を実数係数とする。

分割領域外にあるセルの位置情報は、その時そのセルが持っている領域情報とし、領域情報は4分割を行うごとに更新する。ここで使用する領域情報は、分割領域における中心座標である。概略配置ステップにおいては $f(t)$ の値が小さな個体を優先的に選択する。これは(2)式より仮想配線長の総和が小さく、かつカットライン偏差が小さな個体に相当する。このように多段分割を繰り返すことによって各領域をまたがるカットライン数の差が平滑化され、結果として配線混雑の分散化が図られることになる。

2.3.2 交叉

本論文のコーディングに従えば、セル番号(遺伝子座)と領域(遺伝子)の対に意味があり、遺伝子の順序関係(たとえば、Fig. 6におけるDACD...という順序)は本質的でない。また交叉手法の中で、1点交叉や2点交叉は、遺伝子の順序(順列)が保持されやすい傾向がある。一方、一様交叉は、遺伝子と遺伝子座の対を世代間で保持することと探索領域の広域化という効果が期待できる。そこで、交叉オペレーションとして一様交叉を適用することとした。しかし単純に一様交叉を用いると、新しく生成された個体に対し、セル数を調整することができないという問題が生じる。このセル数の偏りは、結局、分

割の回数が増えることになり、処理速度の低下を引き起こす。

そのため処理の高速化の観点からは、均等な分割(各領域に割り当てられたセル数が等しい)が望ましい分割となる。

そこで均等な分割を行うために、各領域に割り当てられるセル数の上限・下限を設け、交叉によって生成した新たな各個体に対し、ヒューリスティックな調整手法を用いて、各領域に割り当てられたセル数が制限内に収まるようにした。Fig. 9を用いてその調整法を説明する。

Fig. 9 Heuristic adjustment technique

領域ごとのセル数を算出し、セル数の偏りが閾値よりも大きい場合には、最もセルの多い領域から、最も少ない領域に対しセルの移動を行う。Fig. 9では領域Bから領域Cへのセルの移動を行う。

ここで、移動させるセルの選定は、その移動によって仮想配線長が最小になる、すなわち仮想配線長の減分(以後、Decrementと呼ぶ)が最大になるものとする。Decrementの計算方法は、移動元(Area-B)の各セルについて、移動先(Area-C)のセルと接続関係があるならば‘+1’、移動元のセルと接続関係があるならば‘-1’、それ以外の領域は‘0’として計算する。Fig. 9の場合、Cell-2が移動元との接続関係がなく、移動先のセルとの接続関係が二つあるためDecrementが‘2’となり、移動セルとなる。また、セル数の偏りが閾値よりも小さい場合は、同様に、最もセルの多い領域から、最も少ない領域に対しセルの移動を行う。ここでセル数の偏りの閾値は±20%までとした。

2.3.3 突然変異

ここでは、進化の過程(世代)に応じた2種類の突然変異を導入した。2種類の突然変異を選択的(切り替えのタイミングについては4.2で述べる)に用いる目的は、探索の初期の段階では、より大域的な探索を行い、世代が進み探索が最適解の近傍に移るにつれ、その近傍を集中的に探索するためである。

まず、進化があまり進んでいない段階での突然変異について考える。突然変異は、交叉後の個体に対して用いられるため、突然変異前の個体の状態は、各領域に割り当てられたセル数の偏りが調整された後の状態である。そのため、処理速度向上のために突然変異後もこの状態

(各領域に割り当てられたセル数に偏りが無い)を保持するという制限を設ける．そこでこの制限を満たし、かつ遺伝子型を大きく変更する突然変異としてFig. 10 (1)に示すような突然変異(逆位)を用いる．これは二つの遺伝子座をランダムに選び、それらにある遺伝子列を逆順にすることで実現する．これによって多様性のある解を生み出すことができ、探索空間の拡大効果を期待できる．

つぎに、ある程度世代が進むとFig. 10 (2)に示すように、解にわずかな変更を加える突然変異(上書き)を用いる．これは、個体内のあるセルに対して、そのセルが属する領域を変更することで実現する．

Fig. 10 Mutation in outline placement

2.4 概略配置におけるハイブリッド化

概略配置における局所探索では、対象を限定することによって効率のよい探索を、また評価を交換によって生じる仮想配線長の差分として扱うことによって処理時間の短縮を実現する．

具体的には、遺伝子の交換(すなわちセルの交換)を基本とするが、探索効率をあげるために次の制限を設ける．

- ① 同一領域に割り当てられたセルの交換禁止
- ② 同一ネットに属するセルの交換禁止

局所探索によって局所最適解が得られたならば、GAによる進化処理を進めることが望ましい．そこで切り替えのタイミングを決めるために、各ネットの配置領域へまたがる状態に着目する．概略配置では、すべてのネットがFig. 11に示す五つの状態のどれかに当てはまる．

ここでは、処理の簡便化を考慮して、一回のセルの交換で同一領域に収まる可能性のあるネット(状態(1)・(3))に属するセルを局所探索法の対象とする．

つぎに、セルの交換順序を決めるために状態(1)と(3)のネットが局所探索によって改善される利得(以後、Gainと呼ぶ)を比較する．ここで、局所探索法における評価はネットの外接矩形長の1/2とし、各セルは領域の中心にあるものとする．Fig. 12に示すように状態(1)よりも(3)の方がGainが大きい．そこで、Gainの大きい(3)のネットに属するセルを対象に、まず局所探索を行い、つぎに、(1)のネットに属するセルに対し局所探索を行う．

Fig. 13を用いて対象領域をB・Cとした場合の具体的な局所探索の処理手順を示す．ただし、Fig. 13では(1)のようなネットリストがあり、初期状態が(2)である場

Fig. 11 State of nets on outline placement

Fig. 12 Gain by exchange of cells

合と仮定する．

- Step 1 領域B・Cにのみまたがっているネットを選択する．(Fig. 13 (3))
- Step 2 Step 1で選出されたネットで、相反する領域でのペアをつくる．ただし、各ペアのセルは同じネットには属さないものとする．(Fig. 13 (4))
- Step 3 Step 2でつくられた各ペアに対してGainを計算し、Gainが最も大きいペアの交換を行う(Fig. 13 (4)(5))．
- Step 4 Step 1～Step 3をGainが得られなくなるまで繰り返す．

2.5 詳細配置におけるコーディング

詳細配置ではFig. 14に示すように概略配置同様に遺伝子座にCell ID(個々のセルを区別するための番号)を、遺伝子にPlacement Area(配置グリッド名)を対応させる．したがって、詳細配置では、Cell ID順にグリッド名を保存する形で個体を表現する．

2.6 詳細配置における遺伝オペレーション

2.6.1 選択

詳細配置では、個々のセルの詳細な配置位置を決定するため、配線の局所的な混雑緩和を重視した評価関数を導入する．本処理で用いる評価関数の詳細を説明する前

Fig. 13 Hybrid operation in outline placement

Fig. 14 Example of coding in detail placement

に局所的配線混雑度の定義, および算出方法を説明する.

Step 1 一つのネットに着目し, そのネットに属する端子を列挙する.

Step 2 列挙した端子集合において二つの端子のすべての組み合わせをとりだし, それぞれについて二つの端子の最小外接矩形を求める.

Step 3 Step 2で求めた最小外接矩形をチップ上に投影し, 矩形の重なり数を計算する.

Step 4 すべてのネットに対し Step 1~Step 3を行う. Step 3における LSI チップ上への投影は同一チップ上とする.

この重なり数を各配置グリッドにおける局所的配線混雑度とする. また, 局所的配線混雑度の最大値を局所的最大混雑度とする. Fig. 15 (1) の場合, 4 端子ネットであ

Fig. 15 Example of the degree of local wire congestion

るため, Step 2 における矩形は全部で六つ (${}_4C_2=6$) あり, 局所的配線混雑度の分布は Fig. 15 (2) のようになる.

詳細配置で用いた評価関数を (3) 式に示す. 以後, これを “MLC (Measure for Local area wire Congestion)” と呼ぶことにする.

$$f(t) = \left\{ \beta \times \sum_{i=0}^{\text{allNets}} \text{netLength}(i) \right\} + \text{loop} \times \text{maxCongestion} \quad (3)$$

ただし, $\text{netLength}(i)$ は i 番目のネットの外接矩形長の $1/2$ とし, maxCongestion は当該領域における局所的な最大混雑度とする. また, loop は世代数とし, t は個体, β は実数係数とする.

これは, 世代数をパラメータとして, 仮想配線長と局所的な最大混雑度の和をとったものである. 世代数が小さい場合, すなわち GA における世代が進んでいない時は, 仮想配線長を主な評価項目とし, 世代が進むにつれ, 局所的な最大混雑度が評価項目として高い比重を占めるようになる評価関数である. このように MLC では, 最初は仮想配線長が短くなるような配置を行い, 微調整で局所的な配線混雑を緩和する方向に配置を改善していく.

2.6.2 交叉

概略配置では, 一様交叉を用いたが, 詳細配置でこれを用いると一つのグリッドに複数のセルが同時に存在する, すなわちセル同士が重なる致死遺伝子が多く発生する. そこで, 致死遺伝子の抑制を考慮して, 部分一致交叉 (PMX: Partially Mapped Crossover) [8] をベースに本論文でのコーディングに合うように変更を加えた交叉手法を用いた.

2.6.3 突然変異

世代の進捗に応じて 2 種類の突然変異を使用する. これは, 概略配置で導入したのと同じ考えであるが, 前回の近傍探索を行う突然変異 (上書き) をそのまま詳細配置で用いると, その個体はセルが同じ場所に配置された致死遺伝子となる. そこで近傍探索を行う突然変異では致死遺伝子の抑制を考慮して, 任意の二つのセルの位置を交換する突然変異を用いることにした.

2.7 詳細配置におけるハイブリッド化

詳細配置ステップではセルの詳細な位置情報を扱うのに適した局所探索法として FDPR (Force Directed

Pairwise Relaxation) [22]を採用した。FDPRは、セルが互いにバネのような力線で結ばれていると考え、セルに働く張力の合力を最小にする位置を重心と見なしてその位置に配置することを基本とする。これによりネットの外接矩形が小さくなり、配線長が改善されることが期待できる。ここでのFDPRは、Fig. 16に示すようにランダムにネットを選び、そのネットの重心にあるセルと、ネットに属するセルで、交換によって最も評価の良くなるセルとを交換する。しかし、概略配置によって分割された各領域では、ネットが複数領域にまたがる場合がある。そこで、ここでは処理を行っている領域内で接続されているセルが三つ以上あるネットを対象にした。

Fig. 16 Force Directed Pairwise Relaxation

3. 並列計算機への実装

レイアウト設計においてQTAT化は重要な課題である。これを実現する一つの方法として並列処理によって配置処理を高速化させることが考えられる。しかしセル配置問題に対する並列処理方式の報告はこれまであまりなされていない。

一方、GAは多数のデータに対し、独立して並列処理ができることから、今までにも並列化の研究が行われている[17–19]。その一つに、全個体をいくつかの部分個体群に分割し、それぞれの部分個体群にプロセッサ（PU：Processing Unit）を割り当てる並列処理方式がある。これをここでは個体分割並列処理と呼ぶことにする。

この並列処理方式では、各プロセッサが保持する個体数が減ることになり処理の高速化が図られる。Pettyら[17]は、関数最適化問題[21]に適用することによって理論的な解析を行い、その有効性を示している。

ここでセル配置問題の並列化を考えた場合、Fig. 17に示すように各プロセッサにLSIチップの部分領域を割り当て、各プロセッサが割り当てられた部分領域の配置処理を順次行っていく並列化が考えられる。これを領域分割並列処理と呼ぶことにする。

概略配置においては、Fig. 17に示す並列化では処理速度の向上が図れない。なぜなら概略配置では、分割が進むにつれ遺伝子長が短くなるため、1回の分割に要する処理時間も短くなっていく。すなわち最も処理時間を要するのは1回目の分割であり、このような処理時間を多く要する初期の段階の分割において、部分領域が“十分

Fig. 17 Example of parallel processing at cell placement

な数（少なくともプロセッサ数）”生成されていないため、領域分割並列処理が有効であるとはいえない。

そこで、まず概略配置では個体分割並列処理を適用することとし、次に詳細配置では、概略配置によって部分領域が生成されているため、領域分割並列処理を用いることとした。

このように階層処理に適合する二つの並列処理方式を組み合わせることによって、システム全体での高速な配置処理を実現した。

3.1 概略配置における並列処理方式

具体的な概略配置における並列処理方式をFig. 18（四つのPUを1ホスト–3ノードとして使用する）を用いて説明する。まず、各ノードプロセッサ（以後、ノード）で同数の初期個体を発生させ（Step1）、一定世代、各ノードごとで独立に進化をさせる（Step2）。つぎに、各

Fig. 18 Example of parallel processing at outline placement

ノードで一定数の個体をホストプロセッサ（以後、ホスト）を介して別のノードに送る、すなわちノード間で個体の交換を行う（Step3,4）。そして、終了世代まで、各ノードでの独立した進化、および個体の交換を繰り返す（Step5）。このように部分集団を構成する個体を入れ替えることにより集団中の個体の均一化を防ぐ。

個体の交換に関しては、部分個体群の多様性に重点を置き、交換個体の選出はランダムとし、部分個体群のトポロジーはFig. 19に示すようにリング型結合とした。また、Step2の交換頻度およびStep3における交換個体の個数は、いくつかの試行実験（4.4で述べる）を経て決定した。

Fig. 19 Example of ring topology

3.2 詳細配置における並列処理方式

詳細配置での具体的な並列処理方式は、Fig. 20に示すように、部分領域の配置順序に制約がないため、配置処理が終わった待ちノードに順次、部分領域を割り当てることで、同時に複数の領域の配置を行う。これにより処理の高速化を図ることができる。ただし、部分領域の割り当て管理はホストが行っているため、各ノードは部分領域の配置処理ごとにホストと通信を行う必要がある。

Fig. 20 Example of parallel processing at detail placement

ここで、ノードがある部分領域を配置する際、着目したネットでは仮想配線長の見積り精度を上げるには当該領域の外に属するセルの位置情報を把握しておく必要がある。一方、各プロセッサに部分領域を割り当てているため、当該領域外の情報を得るには各ノードで順次更新されていくセルの位置情報をホストからブロードキャ

ストを行わなければならない。しかしこのようなブロードキャストを用いると通信のオーバーヘッドが大きくなる。そこで、セルの位置情報を概略配置終了後の領域の中心座標と近似し、各ノードにあらかじめ持たせておくことによって、通信のオーバーヘッドを低減し、処理速度の向上を図ることとした。

4. 実験評価

4.1 実験要領

本論文で提案した手法の有効性を確認するために、いくつかの比較実験を行った。提案手法では、個体数を200、突然変異率0.01、終了世代を概略配置では、配置セル数回世代交代した後、詳細配置では、配置セル数×10回世代交代した後とした。また、ジェネレーションギャップが0.7の連続世代モデル、トーナメントサイズ（1回のトーナメントで選出される個体数）が4のトーナメント戦略、およびエリート保存戦略を採用した。すなわち毎世代、140の親個体がトーナメント戦略によって選ばれ、次世代の集団を形成する140の子個体を生成する。また、次世代の集団を形成する残りの60の個体（個体数が200のため）の選出はランダムとした。各パラメータについては処理時間と解の精度の点から最も有効な値を予備実験によって求め、設定した。

4.2 チューニング1

まず、選択突然変異の切り替えのタイミングは、評価関数の収束の度合いによって決定した。Fig. 21に概略配置と詳細配置における評価関数の様子を示す。縦軸の評価値は、小さい方が望ましい配置である。Fig. 21より概略配置・詳細配置ともに終了世代の1/6程度の世代が経過した状態で大域的な探索を終了していると読み取ることができる。そこで、選択突然変異の切り替えのタイミングを「（終了世代/6）回世代交代を行った後」とした。

Fig. 21 Relationships between generation and fitness

つぎに、個体群の中で局所探索を適用する個体数を決めるために実験を行った。ここで、局所探索を適用する個体の選出はランダムとし、同世代で同一個体に対し重複して局所探索を適用しないものとした。前提として、ジェネレーションギャップが0.7の連続世代モデルを採

用しているため、毎世代、全個体の30%に相当する個体は遺伝子の組み替えなしに、次世代の個体となる。そのため局所探索を適用する確率を $p\%$ と置いたときに、世代をまたがって同じ個体に局所探索が適用される確率は、(世代 t での局所探索の適用確率 p) $\times$ (遺伝子の組み替えなしに次世代の個体となる確率 0.3) $\times$ (世代 $t+1$ での局所探索の適用確率 p)で、 $0.3 \times p^2$ となる。このことをふまえて、 p が10%, 20%, 50%, 100%について、総仮想配線長を比較したものがFig. 22である。図から $p=20\%$ がすべての場合において最良であることが分かる。 $p=20\%$ の場合では、世代をまたがって同じ個体に局所探索が適用される確率は、 $1.2\%(0.2 \times 0.3 \times 0.2)$ である。一般に、 p が大きい場合には、世代をまたがって同じ個体に局所探索が適用される確率が大きくなり、逆に小さい場合（例えば、適用しない）は局所探索の効果が少ない。そのため、 $p=20\%$ が妥当だと判断し、局所探索法は個体群の中で20%の個体に対し適用するものとした。

Fig. 22 Comparison of the total virtual wire length on the same processing time

4.3 実験結果 1

チューニング1によるパラメータ値を用いて、提案手法 (Hybrid GA)、従来手法、Normal GA の三つの手法の比較実験を行った。従来手法では、広く採用されている手法として初期配置に Mincut 法 [23] を、配置改善に FDPR 法を用いた。また Normal GA では、各個体にランダムな初期配置を与え、交叉では、本論文の詳細配置での交叉を、突然変異には二つの遺伝子をランダムに交換する突然変異を用い、選択オペレーションの評価関数にはネットの最小外接矩形長の1/2の総和を用いた。いずれも実行プラットフォームはワークステーションである。仮想配線長での比較を Table 1 に示す。Table 1 の Average Ratio は、従来手法の配置結果 (仮想配線長) を基準とした場合の Normal GA と Hybrid GA の仮想配線長の相対的な値を各配置規模 (セル数 100, 300, 500, 1000) ごとに算出し、その平均を示している。データサイズにかかわらず、提案手法が最良の結果となっている。これはハイブリッド化によって、探索能力が増す、

Table 1 Comparison of the total virtual wire length

Number of Cells	Mincut +FDPR	Normal GA	Hybrid GA
100	4432	4101	3247
300	39001	42103	28546
500	51601	76267	39668
1000	151682	209639	123051
Average Ratio	1	1.22	0.76

(Unit : channel)

Table 2 Comparison of maximum local wire congestion

Number of Cells	Mincut +FDPR	Normal GA	Hybrid GA
100	43.93	41.54	40.74
300	134.53	146.28	117.91
500	225.14	288.49	199.57
1000	428.75	529.42	391.76
Average Ratio	1	1.14	0.90

(Unit : the number of overlapping rectangles)

すなわち局所探索と大域的探索が効率よく行われたことがその要因と考えられる。つぎに局所的な最大混雑度の比較結果を Table 2 に示す。Table 2 の Average Ratio も Table 1 と同様に、従来手法に対する局所的な最大混雑度の相対比の平均を示している。提案手法では、概略配置で配線混雑の分散化を、また詳細配置では局所的配線混雑の緩和を行う選択関数が有効に機能し、配線混雑緩和が図られた。

4.4 チューニング 2

まず、概略配置での並列化における交換個体の数を決定するために実験を行った。Fig. 23 は、部分個体群の中での交換する個体数を変えた場合の仮想配線長による比

Fig. 23 Comparison of the total virtual wire length on the different number of exchange individuals

較を示す。Fig. 23により、部分個体群の中で、10%の個体を交換する場合が最良の結果となった。

つぎに、交換の頻度を決定するための実験を行った。Fig. 24は、交換個体数を個体群の10%とし、交換の頻度を変えた場合の仮想配線長の比較を示す。Fig. 24では、交換の頻度を終了世代数に対する割合で表しており、2%だと終了世代までに50回の個体の交換を行い、20%だと5回の個体の交換を行う。すなわち、配置セル数が300の場合を例にあげると、終了世代数は300となり、2%の交換頻度だと6回に1回、個体の交換を行うことになる。

Fig. 24 Comparison of the total virtual wire length on the different intervals of migration

Fig. 24より、交換の頻度は終了世代数の5%が最良の結果となった。これは、交換回数が多くなると、各部分個体群での進化の独自性が失われ、逆に少ないと集団としての多様性が失われたと考えられる。

ここで、個体の交換数や頻度が多くなると、交換にかかる通信のコストが増えるが、交換個体数を部分個体群の10%、また、交換の頻度を終了世代数の5%とした時の全体の処理時間における通信処理時間の割合をFig. 25に示す。Fig. 25より、通信に要する処理時間が全体の処理時間の2~4%程度であるため、交換個体数が10%、頻度が5%で問題がないと判断できる。

Fig. 25 Analysis sheet of parallel processing time

4.5 実験結果 2

Fig. 26にチューニング2のパラメータ値を用いて、配置セル数が1000, 2000, 3000の場合の単一PUによるシングル処理と複数PUを用いた並列処理との処理時間の比較を示す。

Fig. 26 Comparison of the processing time on multiple nodes with a single node

Fig. 26より、並列化の理想値に比べると、並列性は若干落ちるが、台数効果による高速化が実現できていると考えられる。Fig. 27に単独処理での仮想配線長を1とした場合の複数PUによる仮想配線長の比較を示す。単独処理に比べ、並列化を行った場合の方が良い結果になったことは、部分個体群における進化の独自性が有効に働いたと考えられる。すなわち、全個体をいくつかの部分個体群に分割することにより、過剰収束（相対的に適応度の高い個体が存在すると競争原理が働き過ぎて、まだ十分に探索が行われていないにもかかわらずその個体によって集団が占拠されてしまう現象）がある程度回避され、部分個体群間で個体の交換を行うことによって、各部分個体群での多様性が維持できたと考えられる。

Fig. 27 Comparison of the total virtual wire length

以上より提案する2種類の並列化手法を組み合わせた並列処理は、配置性能、処理時間の両面において有効な並列処理であるといえる。

最後にネット数 640, セル数 515 のデータを Min-

cut+FDPR を用いて配置した結果と Hybrid GA を用いて配置した結果を示す。Fig. 28 は、配線のエアライン表示（配線すべき二つの端子間を直線で接続したもので、配線混雑の把握に用いられる表示方法）であり、Fig. 29 は局所的配線混雑度の分布を示す。ここで局所的配線混雑度は、山が高いほど、配線が密の部分であり、逆に低い部分は、配線が疎の状態を示す。あきらかに Hybrid GA の優位性を読み取ることができる。

Fig. 28 Placement result (Airline)

Fig. 29 Placement result (Congestion)

5. おわりに

並列計算機に適合したハイブリッド型 GA をベースとする Congestion-Driven 並列配置手法を提案し、いくつかの比較実験によってその有効性を確認した。提案手法は、概略配置において配線混雑の分散化をはかり、詳細配置において局所的な配線の混雑緩和を行うという 2 段階の処理を組み合わせ、広域的な配線混雑の緩和が可能であることを実証した。

また各配置ステップにおいて GA と局所探索法とのハイブリッド探索によって配置性能の向上が図られることを示した。さらに、2 種類の並列化を組み合わせた並列化手法を実現し、従来の単独処理に比べて提案した並列化手法が配置性能、処理時間の両面において有効な並列処理であることを示した。

今後の課題としては、概略配置での並列化における個体の交換の改良があげられる。本論文では、同期式のリング型のトポロジーを持つ並列処理を実現した。しかし理想的には、集団内の個体の多様性が減少した場合のみに、できるかぎり異なった分布を持つ部分集団間のみで個体の交換を行うことが望ましい。そこで、そのような個体交換を実現する交換アルゴリズムを検討中である。

参考文献

- [1] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi: Optimization by simulated annealing; *Science*, Vol. 220, pp. 671–680 (1983)
- [2] J. Holland: *Adaptation in Natural Artificial Systems*, the University of Michigan Press (Second edition, MIT Press) (1992)
- [3] T. N. Bui and B. R. Moon: A fast and stable hybrid genetic algorithm for the ratio-cut partitioning problem on hypergraphs; *Proc. of 31st Design Automation Conference*, pp. 664–669 (1994)
- [4] S. M. Sait, S. Tanvir and M. S. T. Benteen: Timing influenced general-cell genetic floor planner; *Proc. of Asia and South Pacific Design Automation Conference*, A-3A. 3, pp. 135–140 (1995)
- [5] K. Hatta, S. Wakabayashi and T. Koide: 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)
- [6] K. Shahookar and P. Mazumdar: A genetic approach to standard cell placement using meta-genetic parameter optimization; *IEEE Transaction on Computer Aided Design*, Vol. 5, pp. 500–511 (1990)
- [7] 半田, S. Kuga: GA, TS によるアナログ LSI のポリセル方式素子配置; 第 8 回路とシステム軽井沢ワークショップ論文集, pp. 91–96 (1995)
- [8] D. Goldberg and R. Lingle, Jr.: Alleles, loci, and the traveling salesman problem; *Proc. of 1st Int. Conf. on Genetic Algorithms and Their Applications*, pp. 154–159 (1985)
- [9] M. Murakawa, S. Yoshizawa, I. Kajitani, X. Yao, N. Kajihara, M. Iwata and T. Higuchi: The GRD chip: Genetic reconfiguration of DSPs for neural network processing; *IEEE Transaction on Computers*, Vol. 48, No. 6, pp. 628–639 (1999)
- [10] R. M. Brady: Optimization strategies gleaned from biological evolution; *Nature*, Vol. 317, pp. 804–806 (1985)
- [11] D. E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley (1989)
- [12] P. Jog, J. Y. Suh and D. van Gucht: Parallel genetic algorithms applied to the travelling salesman problem; *SIAM J. Optimization*, Vol. 1, pp. 515–129 (1991)
- [13] H. Mühlenbein: Parallel genetic algorithms, population genetics and combinatorial optimization; *Proc. 3rd Int. Conf. on Genetic Algorithms*, pp. 416–421 (1989)
- [14] G. von Laszewski: Intelligent structure operators for the k-way graph partitioning problem; *Proc. 4th Int. Conf. on Genetic Algorithms*, pp. 45–52 (1991)
- [15] 丸山: 遺伝的アルゴリズムを用いた並列グラフ分割アル

- ゴリズム；並列処理シンポジウム (JSPP'92) (1992)
- [16] 城戸：遺伝的アルゴリズムを用いたハイブリッド探索；
遺伝的アルゴリズム (北野編)，pp. 61–86，産業図書
(1993)
- [17] C. C. Pettey and M. R. Leuze: A theoretical investigation of a parallel genetic algorithms; *Proc. 3rd Int. Conf. on Genetic Algorithms*, pp. 398–405 (1989)
- [18] T. C. Belding: The distributed genetic algorithms revisited; *Proc. 6th Int. Conf. on Genetic Algorithms*, pp. 114–121 (1995)
- [19] V. S. Gordon and D. Whitley: Serial and parallel genetic algorithms as function optimizers; *Proc. 5th Int. Conf. on Genetic Algorithms*, pp. 177–183 (1993)
- [20] F. Glover: Tabu search, Part I; *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190–195 (1989)
- [21] K. A. DeJong: An analysis of the behavior of a class of genetic adaptive systems; Ph. D. Thesis, Department of Computer and Communication Sciences, University of Michigan (1975)
- [22] K. Shahookar and P. Mazumdar: VLSI cell placement techniques; *ACM Computing Surveys*, Vol. 6, pp. 143–220 (1991)
- [23] W. Kernighan and S. Lin: An efficient heuristic procedure for partitioning graphs; *Bell System Technical Journal*, Vol. 49, No. 2, pp. 291–307 (1970)
- [24] M. Wang and M. Sarrafzadeh: Modeling and minimization of routing congestion; *Proc. of Asia and South Pacific Design Automation Conference*, D2. 3, pp. 185–190 (2000)
-