

Title	レイアウト階層構造の動的再構築を伴うフロアプラン合成
Author(s)	小原, 正寛
Citation	
Issue Date	2003-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1682
Rights	
Description	Supervisor:金子 峰雄, 情報科学研究科, 修士

修 士 論 文

レイアウト階層構造の動的再構築を伴う
フロアプラン合成

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

小原 正寛

2003 年 3 月

修 士 論 文

レイアウト階層構造の動的再構築を伴う フロアプラン合成

指導教官 金子峰雄 教授

審査委員主査 金子峰雄 教授
審査委員 平石邦彦 助教授
審査委員 浅野哲夫 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

110034 小原 正寛

提出年月: 2003 年 2 月

目次

第1章	はじめに	1
第2章	レイアウト設計について	3
2.1	VLSI チップのレイアウト設計	3
2.2	レイアウトの階層設計	4
第3章	提案手法の概要	5
3.1	問題の定式化	5
3.2	提案手法について	6
3.2.1	Modification and Local refinement(ML 法)	6
3.2.2	Modification and Retry(MR 法)	6
第4章	Force Directed 法に基づいた 配置手法	8
4.1	ブロック配線長の評価	8
4.2	ブロック配置の計算方法	8
4.3	ブロックの重なり除去	9
4.3.1	ブロックの形状変更	11
第5章	階層構造の再構築	13
5.1	ブロックの統合について	13
5.2	ブロックの分割について	15
第6章	実験及び考察	17
第7章	まとめ	32

第1章 はじめに

VLSI(Very Large Scale Integrated circuit)は計算機, 通信機器を始めとする様々な情報処理, 信号処理システムの主要構成要素であり, その製造技術の進歩により伴い, 回路規模の増大および微細化が急速に進んでいる. 年々, その集積度(単位面積当たりの素子数)は増大し, 現在では数百万トランジスタ規模のVLSIチップが出現している. かつて集積度はIC(Integrated circuit)当り数ゲートにすぎなかったが, 最近では計算機のCPU(Central Processing Unit)全体が1チップに集積化されるようになってきた.VLSIの機能はより一層複雑化し, 低消費電力化, 小型化, 高速化等の要求とも相伴い, VLSIの設計は非常に困難な問題となってきた. その設計は工数・期間ともに人手の能力の限界を超えたと言える. このために電子計算機を用いてのVLSIチップの設計を支援, または自動化する, CAD(Computer Aided-Design)の研究は重要な研究テーマとなっている.

チップレイアウト設計に関する研究の歴史は古く, 1961年には既にVLSIチップ上の部品間を配線する基本的な配線アルゴリズムが報告されている. 当時より研究されてきた自動配置配線アルゴリズムは, CADシステムに組み込まれ, 1980年代のその急激なニーズの高まりと共に実用レベルに達してきた. そこでは各種VLSIチップモデル, 各レイアウト設計フェーズに対して自動レイアウトシステムが構築されて, 設計工数削減, 期間短縮に大きく貢献している. しかし, 現在では, 超大型計算機に使用されるVLSIチップ, アナログVLSIチップのレイアウトなど, 単に素子を配置し, 素子間を配線するだけでなく, 回路の電気的特性についても同時に最適化を図る, 等の高度な要求が高まっている. そのためにより広い回路形式とより広い範囲のVLSIチップモデルを対象とし, かつ, 高品質なレイアウトを生成するチップレイアウト設計アルゴリズムの研究が活発に行われている.

レイアウト設計において, 現在では, セル数が数百万に及ぶ回路をチップ上にレイアウトする必要があり, 全てのセルを一括で処理し, 配置を行うことは実用的には困難となっている. これに対し, セルの集合をブロックと呼ばれる部分集合に分割し, (i) 各ブロックの概略位置配置を決定するフロアプラン合成, (ii) 各ブロックにおいて, セルの位置と配線経路を決定するブロック内設計, の二段階で実現している. 従来のこうした階層設計においては, 1) 論理設計の段階で構成された階層構造(論理階層構造)がそのまま用いられるか, 2) 回路分割をグラフの分割問題として捕らえて分割を施す, ことによりブロックは構成されてきた. しかし, J.Congらは最適なレイアウトにおいては論理階層構造がほとんど反映されていないことを示した[1]. このようにレイアウト設計において, 論理階層構造を受け継ぐ必然性がないことが認識されつつあり, レイアウト設計の目的に適合するレイアウトのための階層構造(レイアウト階層構造)を新たに構成する必要がある.

H.Eisenmann と F.M.Johannes は力の概念を配置に導入した Force Directed(FD) 法を提案した [3]. FD 法は配線長の自乗の和の最小化を最適化目標とし, それを実現するために配線をばねに見立て, その力の均衡状態を求めることによりフロアプラン合成を行う手法である.

本研究では入力回路の階層記述にとらわれることなく, 力学的手法を用いたブロックの概略配置と, そこで用いられる力を指標としたブロックの分割, 統合を繰り返し行うことでレイアウト階層構造とフロアプランを同時に実現する. また, これらを実現するための力学的手法に基づくブロックの概略配置とブロックの動的再構築によるフロアプラン合成の位置手法を提案する. これは力学的手法によるブロック (部分回路) 配置の手続き中に回路の部分回路への分割の逐次修正を組み入れることにより, 接続関係だけでなくレイアウトにも配慮した回路分割を行なおうとするものである.

以下の各章では次に示す手順で説明を行う. 第2章では, VLSI 設計行程とレイアウトの設計について簡単に述べる. 第3章では提案手法の概要について説明する. 第4章では, 本研究でも基になっている力学的手法である FD 法について説明する. 第5章では, 本研究で提案するブロックの分割・統合と再配置の手法について述べる. 第6章では, 本手法を計算機上に実装し, 実験を行ったので, その結果について考察し, 7章にてまとめを示す.

第2章 レイアウト設計について

VLSI のチップの設計行程は、大きく論理設計と実装設計に分かれる。論理設計では、システムの設計を行う。この設計では、アーキテクチャの設計、モジュールの設計とその検証を行う。すなわち VLSI 設計の仕様を満たすアーキテクチャレベルの設計である。ここでは、演算ユニット、入出力、メモリなどの各ブロックを取り扱う。これらのブロックは、より詳細なレジスタ転送レベルまで分解される。

次に論理設計を行う。アーキテクチャ設計の結果が論理要素に、すなわち、ゲート、フリップフロップ、マルチプレクサ、等の要素とその結線関係にまで変換される。これを論理レベル表現、または、スキマティックと呼ぶ。論理設計結果は、続く論理検証で検証される。ここでは、論理シュミレーションを実行することによって、論理設計の論理エラー、タイミングエラー、等を発見する。次の診断可能性解析ステップでは、チップ製造時に発生するであろうエラーの観測可能性と制御可能性を解析する。

回路設計において、論理設計結果は、トランジスタレベルの回路に変換される。ここでは、使用する回路形式、例えば、CMOS(Complementary MOS)、ELC(Emitter Coupled Logic)、等を意識してその設計規則、特性を遵守して設計する。回路検証では、回路シュミレータによって回路レベルで所望の論理が実現できているかどうかをチェックする。

レイアウト設計では、論理図と部品情報を入力としてチップのレイアウトパターンを出力する。ここでは、チップ面積と遅延時間等の特性を最適化する。レイアウト検証では、レイアウトにおけるマスクパターンがデザインルールを満足しているかどうか、また、マスクパターン間の寄生効果を抽出して、それらを考慮してもなお回路設計結果が実現されているかどうかを検証する。ここで、デザインルールとは、製造プロセス毎に決まっているマスクパターンの幅、マスクパターンの間隔、等の記述したルールである。これらのレイアウトパターンは、アートワークシステムへの入力とするためにマスクデータに変換される。最後に予め設定した診断率が満足されるまで、診断可能性結果をもとに診断データを生成し、チップ製造後それが論理機能を実現しているか否かの検証に備える。

2.1 VLSIチップのレイアウト設計

VLSI チップのレイアウト設計で採用している階層設計法を説明する。現在では、数百万にも及ぶモジュールを配置しなければならないので、この設計問題の複雑度を考えるとチップ全体を一度に設計することは不可能に近い。そこで、チップを階層的に分割してそれらを独立に、並列的に設計する階層方式を採用している。セルパターン自動生成ではト

ランジスタを配置してトランジスタ間を配線する。ブロックのレイアウトでは、セルをブラックボックスのようにみなし、ブロックを配置してブロック間を配線する。

これらの配置配線問題を解くための処理には、フロアプラン生成、初期配置、配置改善の各処理からなる。また、レイアウト結果をチェックする各種検証プログラムも同時に開発されてきている。これらは、高速化高制度化が課題であるが、フロアプランについては、未だに手動で行われることが多く、有効な自動化が待たれている部分である。

2.2 レイアウトの階層設計

VLSIチップの設計では階層設計を取り入れている。その利点は大規模な設計対象をしようメモリ量、計算時間の点で計算機で取り扱える範囲に分割すること、及び同一階層ないの設計対象を平行して設計できることにある。具体的には、1チップを機能的な単位であるブロックに分割し、さらにブロックをセルに分割する。ここで、ブロックとは、機能的にまとまった論理の集合で、通常、論理設計者が一つの設計単位として扱うものである。ブロック内では、セルを配置し、それらの間を配線する。セルとはフリップフロップ、セクタ、マルチプレクサ、等の数ゲートから数十ゲート程度からなるさらに詳細な機能単位であり、チップ内レイアウトでは、最小単位として取り扱う。一つのセルは数十～数百のトランジスタから構成される。これらの各階層毎に、すなわちブロック配置、ブロック間配線、ブロック内セルは位置配線、セル内トランジスタ配置配線の各プログラムが開発されている。本論文では、VLSIチップ内のブロック配置問題と、これに対する配置配線問題が主題となっており、そのアルゴリズムを提案する。

第3章 提案手法の概要

力の概念を利用したフロアプラン合成は古くから研究されてきた. 本研究では H.Eisenmann と F.M.Johannes が [3] の中で力の概念を配置に導入した Force Directed 法を基にブロックの配置を行う.

FD 法はブロック間を結ぶ配線をばねのようにみなし, ブロックにばねの復元力を働かせ, ブロックを力学的な安定点に移動させることで配置を求める方法である. FD 法ではブロックを大きさを持たない点として扱っており, ブロックがチップの中心付近に集中した, 重なりを多く含む配置を出力するため, 後行程でブロックを大きく動かし, この重なりを除去しなくてはならない. [3] ではブロックを拡散させる力を加えることで, ブロックのチップ中心付近への集中を緩和させている. また [4] では, ブロックの重なり面積に比例する力を加えることで, 同じく, ブロックのチップ中心付近への集中を緩和させている.

次に FD 法によって得られたブロック配置を基に回路階層構造の再構築を行う. 配置均衡時における力の状況に着目して, ブロックの統合, 分割を行う手法を考案する. これにより各ブロックの概略位置が求まることとなる. 以上のようにブロック配置と回路階層構造の修正を動的に繰り返すことで, 最適なレイアウト階層構造とそれに対するフロアプランを同時に得ることを目的とする.

3.1 問題の定式化

本研究で対象とする問題は以下に示す通りである.
入力

- セルの集合 $S = \{s_1, s_2 \dots s_l\}$

セルは配置位置が予め与えられた固定セル (外部端子など) と, 与えられていない可動セルとに分けられる. また, それぞれ幅, 高さが与えられる.

- セルの接続関係

- S の初期分割 $S_1^0, S_2^0 \dots S_n^0$ (但し, $\bigcup_{i=1} S_i = S, S_i \cap S_j = \{\phi\} (i \neq j \text{ のとき})$)

ここで, 一つのセルは単独で一つのブロックを構成するものとする. このブロックを位置固定ブロックと呼ぶ. 一方, 可動セルだけからなるブロックを可動ブロックと呼ぶことにする.

一つのネット $n_k \in N \subseteq 2^S \setminus \{\phi\}$ は配線にて連結すべきセルの集まりをあらわす.

- 各ブロックの面積の上限 A_B とブロックの総数の上限 N_B
- 配置領域の幅 W , 高さ H

である.

出力

- S の分割 $S_1, S_2 \dots S_n$ (但し, $\bigcup_{i=1} S_i = S, S_i \cap S_j = \{\phi\} (i \neq j \text{ のとき})$)
- ブロックの配置位置である.

最適化目標は以下の条件を満たすようなブロック間の配線長の自乗の和の最小化である.

1. ブロック S_i の面積 $\leq A_B$
2. ブロックの総数 $n \leq N_B$
3. ブロックの重なりのない配置

3.2 提案手法について

以上のような問題に対して本研究では, 以下の2通りの手法を考案し, その有効性について実験を通して検討を行なった. ここで, その手法について簡単に説明する.

3.2.1 Modification and Local refinement (ML 法)

Modification and Local refinement (以下, ML 法) では次のようにしてブロック階層構造の変更, またはブロック配置の下でのブロック間の配線長の自乗の和を最小化を行なう.

1. ブロックの大きさを無視した質点配置
2. ブロックの移動によるブロック間の重なり除去
3. ブロックの統合または分割
4. 3 で構成された新しいブロックを統合/分割前の元のブロックの場所に置き, (2) へ戻る

この手法では, 一つのブロックが統合/分割されると, 前回の階層構造の変更における他のブロック配置位置を考慮して, 統合/分割されたブロックを配置することで, 局所的な最適化を図る手法である.

3.2.2 Modification and Retry (MR 法)

Modification and Retry (以下, MR 法) では次のようにしてブロック階層構造の変更, またはブロック配置の下でのブロック間の配線長の自乗の和を最小化を行なう.

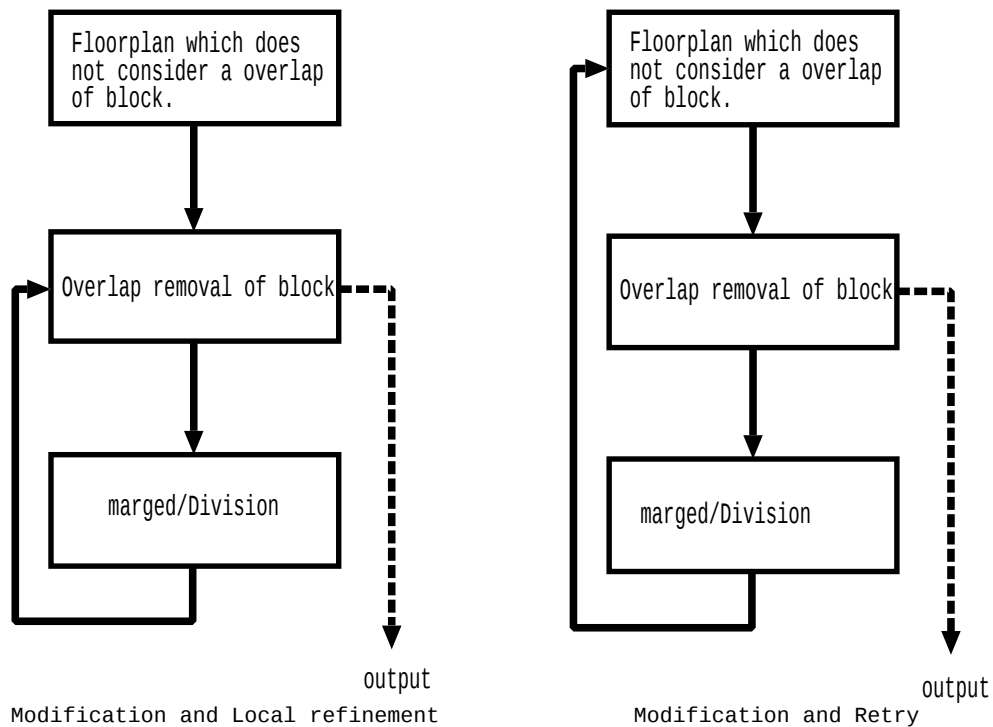


図 3.1: Modification and Retry と Modification and Local refinement

1. ブロックの大きさを無視した質点配置
2. ブロックの移動によるブロック間の重なり除去
3. ブロックの統合または分割
4. 1に戻る

この手法では, ブロック階層構造の変更が起こるたびにブロックの大きさを無視した最適な位置へと書き戻されることになる. その最適な位置から, ブロックの重なり除去を行なう.

第4章 Force Directed法に基づいた 配置手法

4.1 ブロック配線長の評価

ブロック i, j 間の配線のコストを, i, j のユークリッド距離の自乗に i, j 間のネット重みをかけた値とする [2,3,4]. すなわち i, j の座標をそれぞれ $(x_i, y_i), (x_j, y_j)$, i, j 間のネットの重みを C_{ij} とするとき, コストは $C_{ij}\{(x_i - x_j)^2 + (y_i - y_j)^2\}$ となる.

ここでブロック間のネットの重みについて補足する. ここでブロック S_i, S_j 間のネットの重みは C_{ij} は, S_i, S_j を接続する部分ネットリストを $N_{ji}, n_k \in N_{ij}$ が接続するブロック数を d_k として, 次のように定義する.

$$C_{ij} = \sum_{n_k \in N_{ij}} \frac{1}{2d_k}$$

自乗配線長のコストは配線長を忠実に反映する訳ではないが, 変数に関して連続微分可能で数学的に取り扱い易いこと, 及び長い配線を抑制する効果があることから, しばしば採用される評価である.

配線の自乗の和によるコスト $W_n(\vec{p}_x, \vec{p}_y)$ は, 全てのブロックの組のコストの和とする. すなわち

$$W_n(\vec{p}_x, \vec{p}_y) = \frac{1}{2} \sum_{\forall i, j (i \neq j)} C_{ij} \{(x_i - x_j)^2 + (y_i - y_j)^2\}$$

$\vec{p}_x = (x_1, x_2, \dots, x_n)^T, \vec{p}_y = (y_1, y_2, \dots, y_n)^T$ である.

4.2 ブロック配置の計算方法

ブロックの配置を行なうには以下に示すような計算を用いる.

自乗配線長のコスト $W_n(\vec{p}_x, \vec{p}_y)$ は x 方向成分と y 方向成分とに分離可能であることが

ら,以降,記述の簡単化のため, x 成分だけを考慮して変形する.

$$\begin{aligned}
W_n(\vec{p}_x) &= \frac{1}{2} \sum_i \sum_j C_{ij} (x_i - x_j)^2 \\
&= \sum_i x_i \left(\sum_j \tilde{C}_{ij} x_j \right) \\
&= \vec{p}_x^T \tilde{C} \vec{p}_x
\end{aligned} \tag{4.1}$$

ここで $\vec{p}_x$ の x_1 から x_m までを可動ブロックの中心座標, x_{m+1} から x_n までを位置固定ブロックの中心座標とし, それぞれを $\vec{p}_{xM}, \vec{p}_{xF}$ とする座標ベクトルすると $\vec{p}_x$ は $\vec{p}_x = (\vec{p}_{xM}, \vec{p}_{xF})^T$ となる. それに対応して $\tilde{C}$ は

$$\tilde{C} = \begin{pmatrix} C_{MM} & C_{MF} \\ C_{FM} & C_{FF} \end{pmatrix}$$

よってブロック配置のコスト関数 W_n は, 行列を用いて次のように表すことができる.

$$\begin{aligned}
W_n(\vec{p}_x) &= \vec{p}_x^T \tilde{C} \vec{p}_x \\
&= (\vec{p}_{xM}^T C_{MM} + \vec{p}_{xF}^T C_{FM}) \vec{p}_{xM} \\
&\quad + (\vec{p}_{xM}^T C_{MF} + \vec{p}_{xF}^T C_{FF}) \vec{p}_{xF} \\
&= \vec{p}_{xM}^T C_{MM} \vec{p}_{xM} + \vec{d}_x^T \vec{p}_{xM} + const.
\end{aligned}$$

C_{MM} は m 次元正方行列で, ブロックの接続関係より決定する. $\vec{d}_{xM}, \vec{d}_{yM}$ は m 次元ベクトルで, 固定ブロックとの接続から決定する.

W_n は $\forall x_i, y_j$ に対する正の2次式なので, 偏微分係数が同時に0になる時, W_n は最小になる. これは, ネットをばねにモデル化して均衡状態を計算するのに相当する.

$$\begin{pmatrix} \frac{\partial W_n}{\partial x_1} \\ \vdots \\ \frac{\partial W_n}{\partial x_m} \end{pmatrix} = \left(C_{MM} \vec{p}_{xM} + \vec{d}_{xM} \right) = \vec{0} \tag{4.2}$$

y 成分についても同様にして $C_{MM} \vec{p}_{yM} + \vec{d}_{yM} = \vec{0}$ を得る.

4.3 ブロックの重なり除去

次に重なりを除去するために, 上の式を次のように変更する [3,4].

$$\begin{pmatrix} C \vec{p}_{xM} + \vec{d}_{xM} + \vec{e}_x \\ C \vec{p}_{yM} + \vec{d}_{yM} + \vec{e}_y \end{pmatrix} = \vec{0} \tag{4.3}$$

$\vec{e}_x, \vec{e}_y$ はブロックの間の反発力に相当し, 該当するブロックを遠ざける. $\vec{e}_x, \vec{e}_y$ は次のように計算する. まず, ブロック i がブロック j から受ける反発力の x 成分 $f_x(j \rightarrow i), y$ 成分 $f_y(j \rightarrow i)$ を計算する.

この力については以下のように考える. ブロックには重なっている面積に比例する力が働いているものとし, その値を $f(j \rightarrow i) = K \cdot X \cdot Y$ とする (K は定数). f の x 成分, y 成分は圧力がかかっているように考えて, それぞれ $X(i, j), Y(i, j)$ の床があるとする. そこに一定の力 α を受けているものとする. そこで,

$$f_x = \alpha Y$$

$$f_y = \alpha X$$

と仮定する. そうすると

$$\sqrt{f_x^2 + f_y^2} = K \cdot X \cdot Y \quad (4.4)$$

となり,

$$\alpha = \frac{K \cdot X \cdot Y}{\sqrt{X^2 + Y^2}}$$

が得られる. よって $f_x(j \rightarrow i), f_y(j \rightarrow i)$ は

$$\begin{aligned} f_x(j \rightarrow i) &= \frac{K \cdot X(i, j) \cdot Y(i, j)^2}{\sqrt{X(i, j)^2 + Y(i, j)^2}} \\ f_y(j \rightarrow i) &= \frac{K \cdot X(i, j)^2 \cdot Y(i, j)}{\sqrt{X(i, j)^2 + Y(i, j)^2}} \end{aligned} \quad (4.5)$$

これを全てのブロックの組に対して求め, $\vec{e}_x, \vec{e}_y$ とする.

$$\begin{aligned} \vec{e}_x(i) &= \sum_{\forall j(\neq i)} f_x(j \rightarrow i) \\ \vec{e}_y(i) &= \sum_{\forall j(\neq i)} f_y(j \rightarrow i) \end{aligned} \quad (4.6)$$

以上より, 現在の可動ブロックの位置が $\vec{p}_{xM}^i, \vec{p}_{yM}^i$ のとき, 次の可動ブロックの位置 $\vec{p}_{xM}^{i+1}, \vec{p}_{yM}^{i+1}$ を次の様に求める. まず, 現在の位置に対し, 式 (4.6) から, 各ブロックに対する反発力の和が求められる. それを用いて, 式 (4.3) より, $\vec{p}_{xM}, \vec{p}_{yM}$ を求める.

以上より, 次の可動ブロックの位置は,

$$\begin{aligned} \vec{p}_{xM}^{i+1} &= (1 - \alpha) \vec{p}_{xM}^i + \alpha \vec{p}_{xM} \\ \vec{p}_{yM}^{i+1} &= (1 - \alpha) \vec{p}_{yM}^i + \alpha \vec{p}_{yM} \end{aligned}$$

である. ここで, α は 0 以上 1 以下の定数である.

この操作を K が小さい値から大きな値になるまで変更しながら, 一つの K の値において, 全てのブロックの重なり面積の合計が十分小さくなるまで繰り返すこととした.

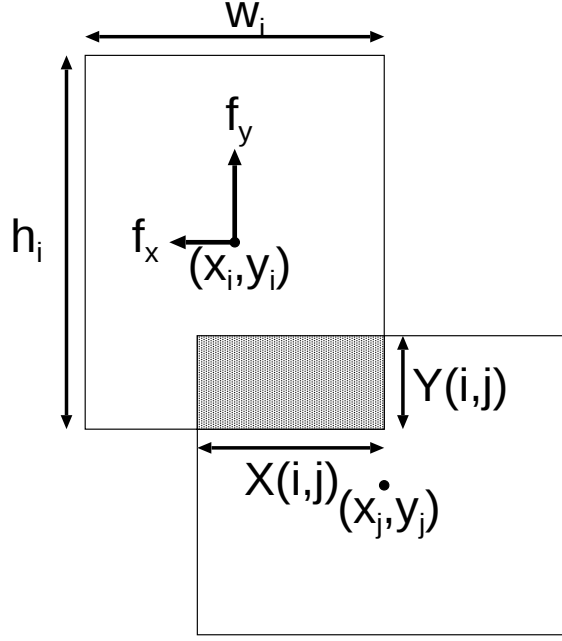


図 4.1: ブロックの重なりによる力

4.3.1 ブロックの形状変更

これまで述べてきたように, ブロック間の重なりを除去するため, 反発力を導入したが, 面積効率を高めるためには, ブロックの形状も考慮する必要がある. まず形状を変更するブロックを次のようにして決定することとした.

式 (4.5) で求めた 2 つのブロック S_i, S_j 間の力の x 成分 $f_x(j \rightarrow i)$, y 成分 $f_y(j \rightarrow i)$ のついて, 力の向きを考える. ブロック S_i の中心座標が S_j より右側にあれば x 軸方向で + の力, 左側にあれば - 方向の力とする. 同様に, ブロック S_i の中心座標が S_j より上側にあれば y 軸方向で + の力, 下側にあれば - 方向の力とする. これを全てのブロックに対して求める.

その後, それぞれの方向に対する力の和により, 各ブロックの形状を以下の様に決定する. 今, ブロック S_i の幅が W_i , 高さが H_i とする.

$$(\min(f_y) > \max(f_x) \times t_A) \wedge (\max(f_y) < t_B \times \min(f_y))$$

を満たすとき,

$$\begin{aligned} W_{i+1} &= \beta W_i, \\ H_{i+1} &= \frac{W_i \times H_i}{W_{i+1}}. \end{aligned}$$

逆に

$$(\min(f_x) > \max(f_y) \times t_A) \wedge (\max(f_x) < t_B \times \min(f_x))$$

を満たすとき,

$$\begin{aligned} H_{i+1} &= \beta H_i, \\ W_{i+1} &= \frac{W_i \times H_i}{H_{i+1}}. \end{aligned}$$

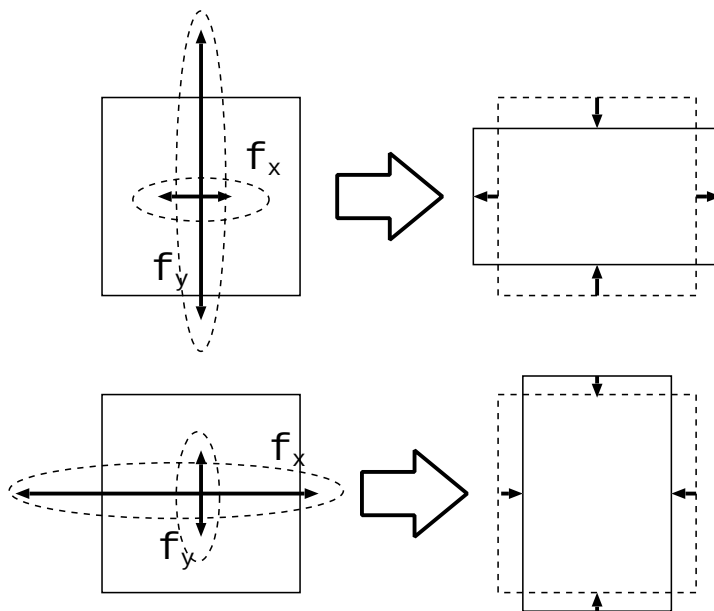


図 4.2: ブロック形状の変更

この操作により、ブロック間に隙間無く、さらに重なりの少ない配置位置を求める。形状変更の操作は、重なり除去のパラメータ K がある程度大きくなったところから、定数回毎に行なうものとする。

第5章 階層構造の再構築

統合・分割に関しては k 個のブロックを統合して l 個ブロックに分割することを考える。ただし、

$$l \leq N_B - n_B + k$$

である。今回は $k \leq 2, l \leq 2$ の場合について行う。

なお、統合と分割をしたあとに、FD 法によって再び配置を行い、これを繰り返すことで最終的な出力を得ようとするものである。現在、得られている配置を基に階層構造の再構築を行う。本研究では、2つのブロックの統合と1つのブロックの2つのブロックへの分割を繰り返し適用することとした。

5.1 ブロックの統合について

ブロックの統合では、まず、統合すべき2個のブロックを選択しなければならない。そこで本研究では、以下に示す2通りにて統合対象ブロックを選択した。

1. 統合対象ブロック選択法 1

- (a) 中心間距離がある定数以下
- (b) ブロック面積の和がブロックの面積制約満たす

以上の場合において、条件を満足するモジュール対の内から、連結度と中心座標間距離を考慮した評価関数に基づいて統合するモジュール対を選択する

2. 統合対象ブロック選択法 2

- (a) 中心間距離がある定数以下

以上の場合において、条件を満足するモジュール対の内から、連結度と中心座標間距離を考慮した評価関数に基づいて統合するモジュール対を選択する

つまりは選択法 1 においてはブロックの統合を行う前に統合後のブロックの面積がブロックの面積の制約を越えていた場合は統合を行わないが、選択法 2 では統合後のブロックの面積がブロックの面積の制約を越えていた場合でも統合することを許し、後の分割による工程で新たにブロックの再構築を図ろうとしたものである。

具体的には、まず始めに、各ブロックに対して隣接するほかのブロックとの統合後の面積を計算する。ブロックの統合後の面積が面積の制約条件を満たし、かつブロックの中心間距離がある閾値以下であるものを統合の候補とする。ここで、中心間距離の閾値として

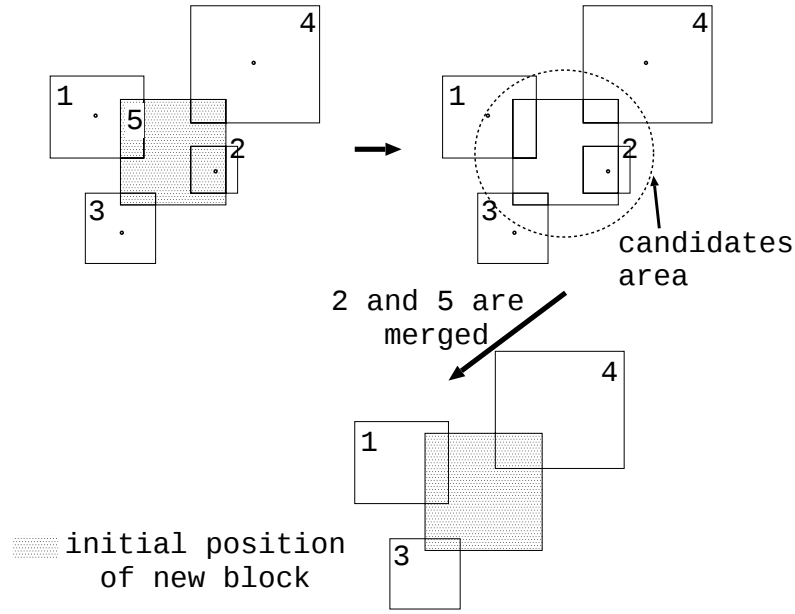


図 5.1: ブロックの統合

は、面積制約と面積が一致した形状が正方形であるブロックの対角線の長さの半分、すなわち、 $\frac{\sqrt{2A_B}}{2}$ を採用した。

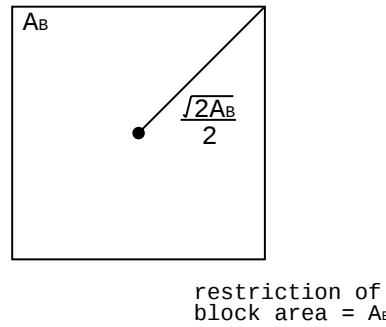


図 5.2: 統合における中心間の閾値

この中心間距離の閾値を考えることにより、中心間距離が大きいブロック対の統合を避けることができ、統合した際の他のブロックへの影響が小さくできることが期待される。また、面積の和を考慮することにより、制約を満たす回路分割が実現される。

これら統合対象のブロック対の候補の中で、関数

$$\begin{aligned}
 f_B &= \frac{\text{2つのブロック間の連結度}}{\text{2つのブロックの中心間距離}} \\
 &= \frac{C_{ij}}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}
 \end{aligned} \tag{5.1}$$

の値が最大となるようなブロック対を統合対象として選ぶ。この関数は、中心間距離が近く、結線数が大きいブロック対に対し、大きな値を返す。

本研究においては、この統合操作によってできたブロックはある一定回の間は、分割対象としない。これは、同一のブロックの統合、分割が繰り返されることを抑制するためである。

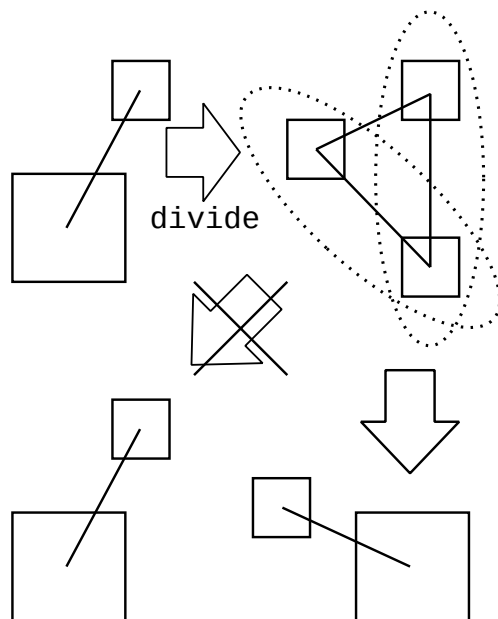


図 5.3: 禁止制約の下での統合対象ブロックの選択

ML 法においては、統合によって新たにできたブロックの中心座標は、統合対象のブロック対で面積が大きい方のブロックの中心座標とする。

5.2 ブロックの分割について

ブロックの分割対象の選択は以下の通りである。まず、ブロックの面積制約を満していないブロックが存在するならば、そのブロックを分割対象とする。もし、そのようなブロックが存在していないならば、以下の2通りの方法によって分割対象ブロックを決定する。

1. 分割対象ブロック選択法 1

各ブロックでFD法におけるブロック間配線による力の合計値を求め、その値が最大であるようなブロックを分割の対象とする。この合計値は、配線によって他のブロックから引かれている力が最も大きいので、よって、そのブロックを分割することにより、配線長の改善につながることを期待される。

2. 分割対象ブロック選択法 2

各ブロックでFD法におけるブロック間配線による力の水平、垂直方向成分の正負の向きの大きさをそれぞれ求め、次に水平、垂直方向それぞれで正負の向きの成分の大きさの積を求める。そして、この積の内、大きい方の値が最大であるようなブロックを分割の対象とする。この積が大きいことは、そのブロックが配線によって水平、もしくは垂直方向での相対する方向から引かれていることを意味し、よって、そのブロックを分割をすることにより、配線長の改善につながることを期待される。

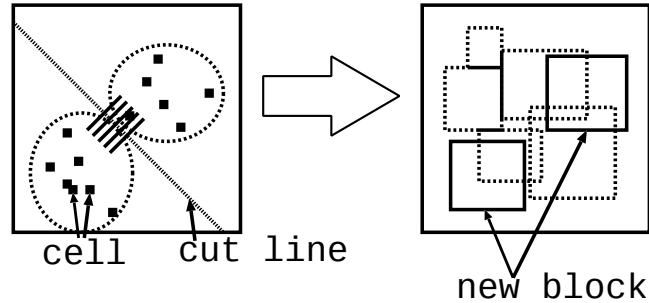


図 5.4: セルの質点配置によるカットラインの選択法

次に分割対象となったブロックに対し、i) 対象ブロック以外のブロックの位置を固定し、対象ブロックに含まれるセルの質点配置を求める。ii) 垂直、水平、もしくは斜め45度の直線(カットライン)によってカットされる配線数(カット数)を求め、そのカット数が最小となるカットラインで質点配置されたブロック内セルの集合を分割し、それを新しい分割として採用する。ここで、カット数最小のカットラインは一番端に存在することが多いことが予想される。そこで、カット数の極大と極大の間にあるような極小の中でカット数最小なカットラインを採用する。

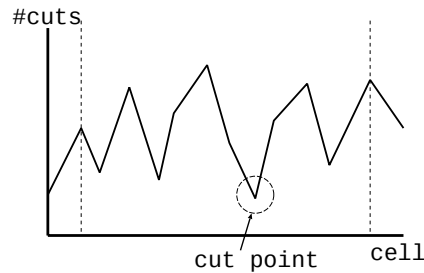


図 5.5: カットラインの移動によるカット数の変化の模式図と分割カットラインの選択

ML法においては、分割した後の2つのブロックの中心座標は、分割する前のブロックの中心座標とする。また、統合の時と同様、この分割操作によってできたブロック対はある一定回の間は、統合対象としない。

第6章 実験及び考察

この章では ML 法, MR 法における実験結果について報告し, 最後に考察を加える.

ここではセル数 149, ネット数 146 の回路, ブロックの面積制約 A_B を 8.4×10^5 , ブロックの総数の制約 N_B を 20(初期は 5) とした入力 (fract) とセル数 914, ネット数 984 の回路, ブロックの面積制約 2.7×10^6 , ブロックの個数の制約 40(初期は 14) とした入力 (primary1) に対して, ML 法, MR 法にて各手法, 統合対象ブロック選択法と分割対象ブロック選択法 (以下は統合対象ブロック選択法 (1)(2), 分割対象ブロック選択法 (1)(2) を単に統合 (1)(2), 分割 (1)(2) とする) をそれぞれ 2 通りの組み合わせで, ML 法, MR 法の各手法計 4 通りの実験を行なった.

なお, 今回の初期分割はカット数最小化を目的としたグラフ分割を基に生成したものである.

ここで実験に用いたパラメータを示す. fract, primary1 共通の値として $t_A = 4.0$, $t_B = 1.5$, $\beta = 0.05$ とし, K の値は 0.2 から 8.0 まで 0.2 ずつ変更させた. また, fract のチップサイズを 2000×2000 , α を 2.5×10^{-4} , primary1 のチップサイズを 6000×6000 , α を 5.0×10^{-5} とした.

初期配置における自乗配線長については, fract は 4.43×10^7 , primary1 は 8.97×10^8 である.

以下において種々の比較結果を示す.

- ML 法, MR 法での実験結果

統合ブロック選択法 1, 分割ブロック選択法 1 の場合

統合ブロック選択法 1, 分割ブロック選択法 2 の場合

統合ブロック選択法 2, 分割ブロック選択法 1 の場合

統合ブロック選択法 1, 分割ブロック選択法 2 の場合

ML 法では, 初期配置からの改善がほとんど見られなかった. 一方, MR 法では, fract に対し約 21.0 %, primary1 に対し約 9.9 % の改善が確認できた. これは本稿で採用した回路階層構造の変更手法が局所的ではなかったため, ML 法の局所的な最適化に適合せず, MR 法のような全体を変更する必要があったためと考察される. また ML 法は一度, 配線長が増大するような階層構造の変更が行われると配線にとって悪い配置のまま, 次の階層構造の変更を行うので, 再び, 配線長の改善に至ることができなかった. しかし MR 法は ML 法の約 2 倍程度, 実行時間を要する. よって, ML 法に適した回路再構成手法の開発が必要である.

統合分割禁止制約により, 同じブロックが統合や分割の対象として選ばれることを防ぎ,

表 6.1: fract に対する実験結果 (統合 (1)/分割 (1))

	統合・分割禁止制約		
	1 回	3 回	5 回
<i>ML</i> 法	4.150×10^7	3.999×10^7	4.002×10^7
<i>MR</i> 法	3.672×10^7	3.755×10^7	4.003×10^7

表 6.2: primary1 に対する実験結果 (統合 (1)/分割 (1))

	統合・分割禁止制約		
	5 回	10 回	15 回
<i>ML</i> 法	8.889×10^8	8.889×10^8	8.889×10^8
<i>MR</i> 法	8.872×10^8	$8,701 \times 10^8$	7.958×10^8

その結果, 回路階層構造の多様性が導かれる. 表 6.11 などでおこる配線長の振動状態を抑制する意味では, 統合分割禁止制約の有効性を確認できた. また, これにより探索が局所最適解に陥らないことが期待される. その一方, 配線長を増やしてしまうようなブロック統合/分割が頻繁に選ばれる可能性もあるため, 結果として配線長が改善されない場合も実験結果より言えるので, よって良質な解を得るためには, 入力回路に適切な禁止制約を定める必要がある.

また, 本研究で採用した統合対象ブロックの選び方は直後で分割対象ブロックとして選ばれるようなブロック対でも統合を行ってしまう. このような統合が行われると, 分割禁止制約により分割を行わないのであまり配線の力が加わっていないようなブロックを選び分割をする, というように階層構造の変更が配線長にとって悪くなる方向に働く可能性がある. *MR* 法においては階層構造が変更される度に質点配置を行うので再び, 配線長の改善に向かう可能性があるが, *ML* 法では前の状態の階層構造を考慮しているので, そのまま配線長が増大することが多かった. 本研究では統合/分割対象ブロックの選択法についてそれぞれ 2 手法ずつ提案したがより配線長が改善されるような統合/分割対象ブロックを決定するアルゴリズムの開発が必要である.

本研究では階層構造の変更回数を定数回で終了としたが, 配線長の改善が一定回起こらなかった場合はそこで探索を終了するなどアルゴリズムの終了条件の判定も今後は必要である.

表 6.3: fract に対する実験結果 (統合 (1)/分割 (2))

	統合・分割禁止制約		
	1 回	3 回	6 回
<i>ML</i> 法	4.099×10^7	4.141×10^7	4.141×10^7
<i>ML</i> 法	3.599×10^7	3.502×10^7	3.851×10^7

表 6.4: primary1 に対する実験結果 (統合 (1)/分割 (2))

	統合・分割禁止制約		
	5 回	10 回	15 回
<i>ML</i> 法	8.889×10^8	8.889×10^8	8.889×10^8
<i>MR</i> 法	8.975×10^8	8.707×10^8	8.097×10^8

表 6.5: fract に対する実験結果 (統合 (2)/分割 (1))

	統合・分割禁止制約		
	1 回	3 回	5 回
<i>ML</i> 法	4.151×10^7	4.113×10^7	4.113×10^7
<i>MR</i> 法	4.345×10^7	3.639×10^7	3.444×10^7

表 6.6: primary1 に対する実験結果 (統合 (2)/分割 (1))

	統合・分割禁止制約		
	5 回	10 回	15 回
<i>ML</i> 法	8.889×10^8	8.889×10^8	8.889×10^8
<i>MR</i> 法	8.975×10^8	8.975×10^8	8.975×10^8

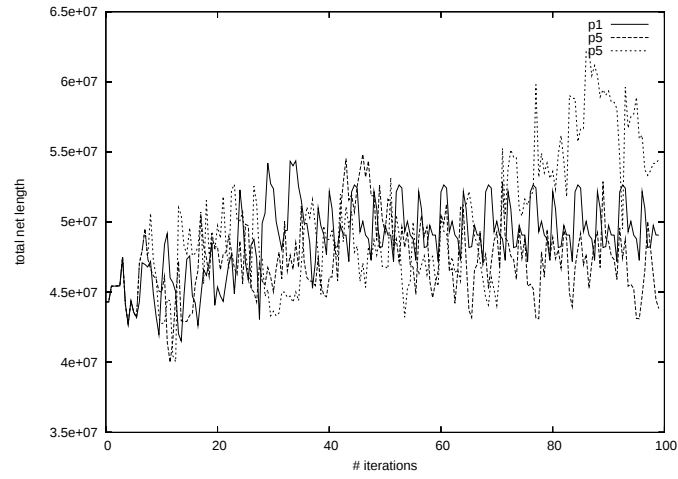


図 6.1: fract に対する ML 法による実験結果 (統合 (1)/分割 (1) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

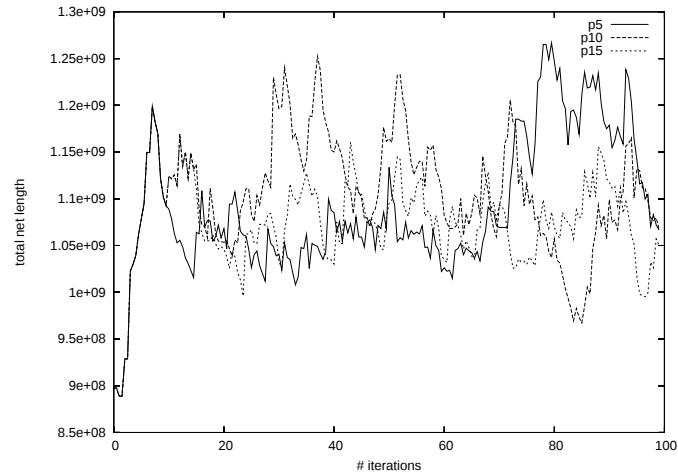


図 6.2: primary1 に対する ML 法による実験結果 (統合 (1)/分割 (1) の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 10 回, p15 は禁止制約 15 回の結果)

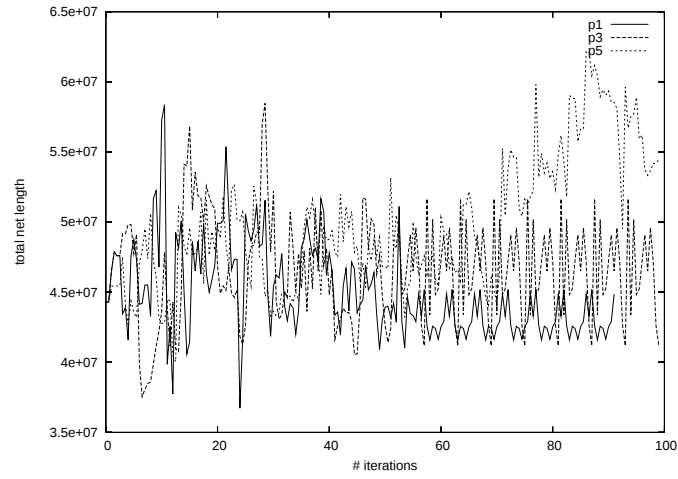


図 6.3: fract に対する MR 法による実験結果 (統合 (1)/分割 (1) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

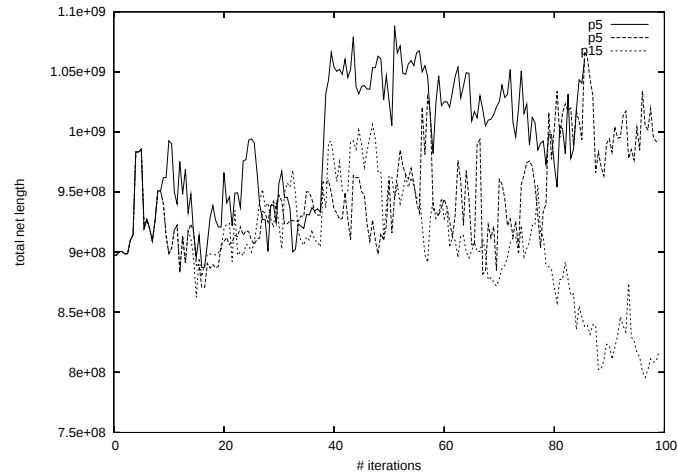


図 6.4: primary1 に対する MR 法による実験結果 (統合 (1)/分割 (1) の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 10 回, p15 は禁止制約 15 回の結果)

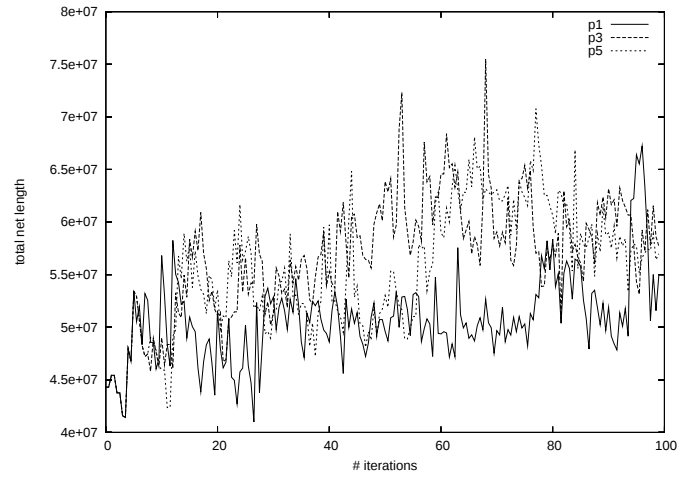


図 6.5: fract に対する ML 法による実験結果 (統合 (1)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

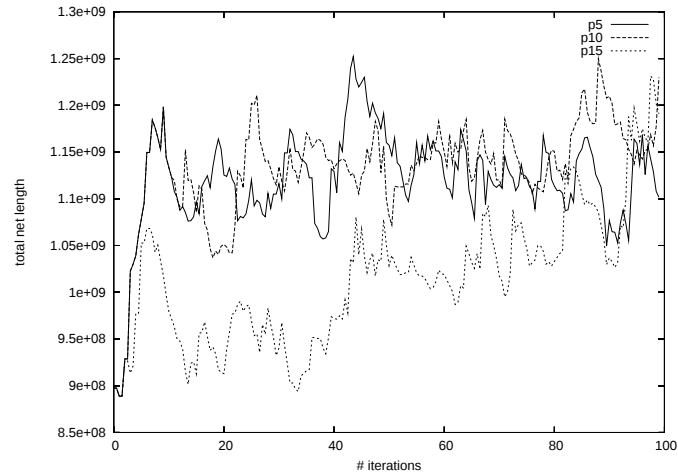


図 6.6: primary1 に対する ML 法による実験結果 (統合 (1)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 10 回, p15 は禁止制約 15 回の結果)

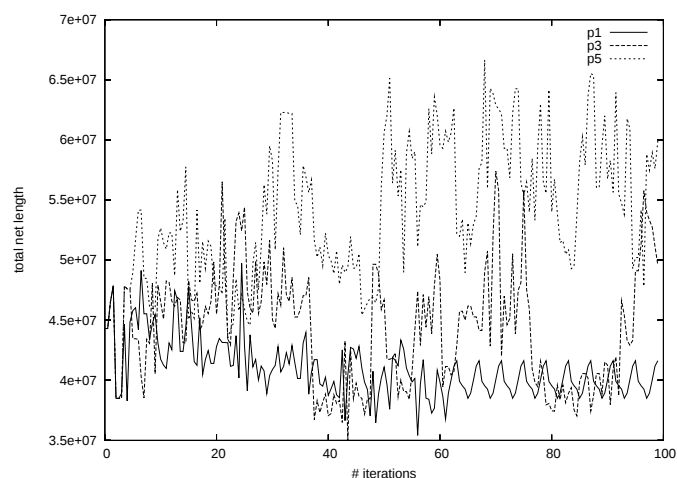


図 6.7: fract に対する MR 法による実験結果 (統合 (1)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

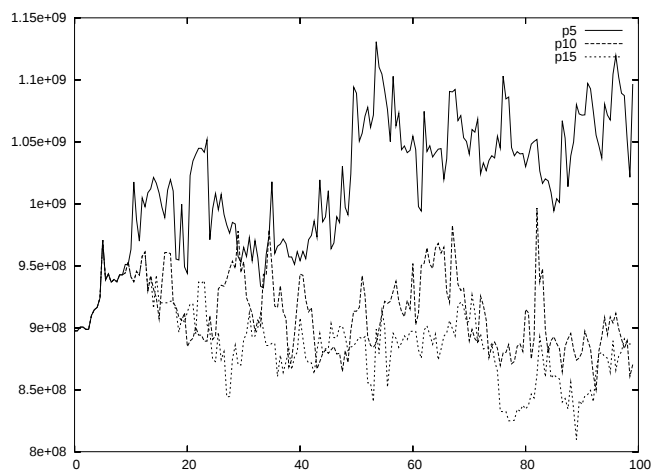


図 6.8: primary1 に対する MR 法による実験結果 (統合 (1)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 10 回, p15 は禁止制約 15 回の結果)

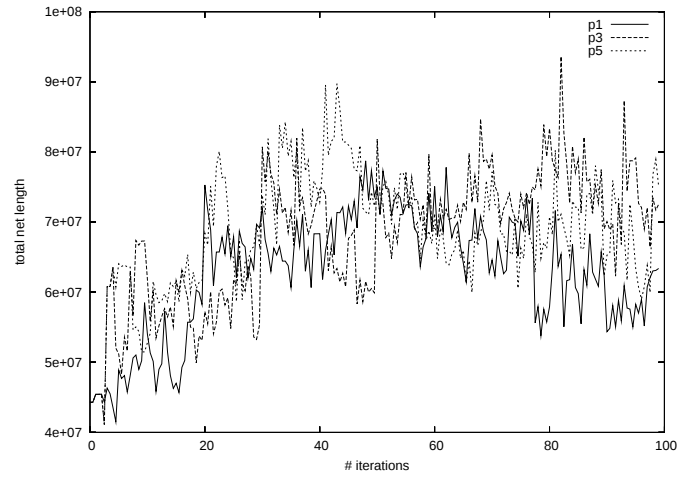


図 6.9: fract に対する ML 法による実験結果 (統合 (2)/分割 (1) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

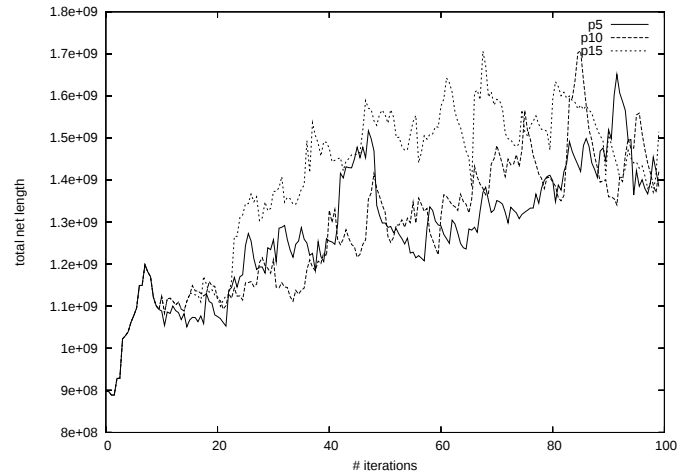


図 6.10: primary1 に対する ML 法による実験結果 (統合 (2)/分割 (1) の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 3 回, p15 は禁止制約 15 回の結果)

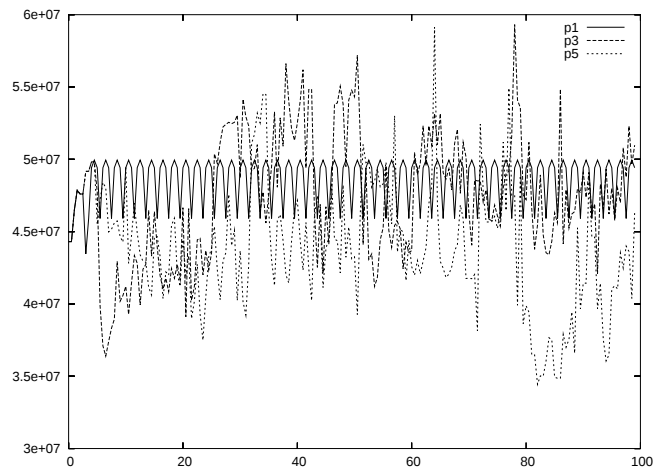


図 6.11: fract に対する MR 法による実験結果 (統合 (2)/分割 (1) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

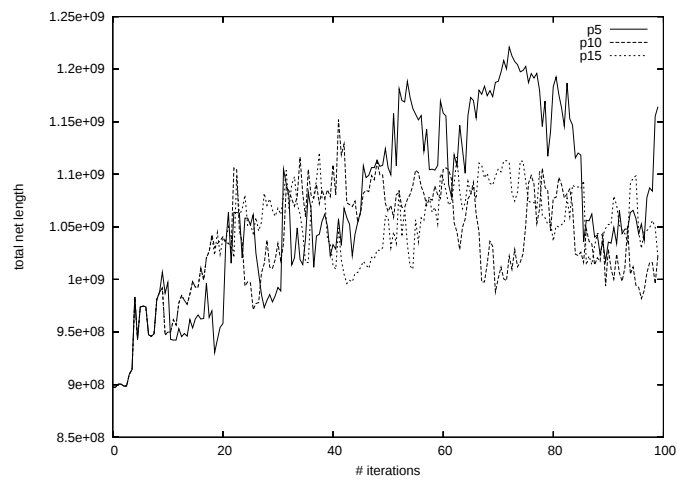


図 6.12: primary1 に対する MR 法による実験結果 (統合/分割の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 3 回, p15 は禁止制約 15 回の結果)

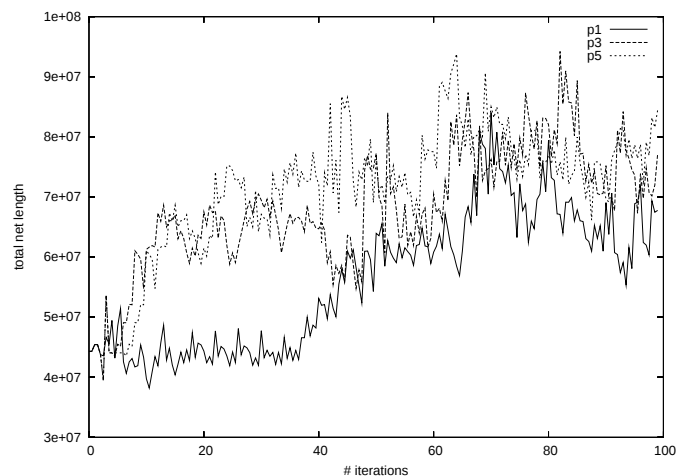


図 6.13: fract に対する ML 法による実験結果 (統合 (2)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

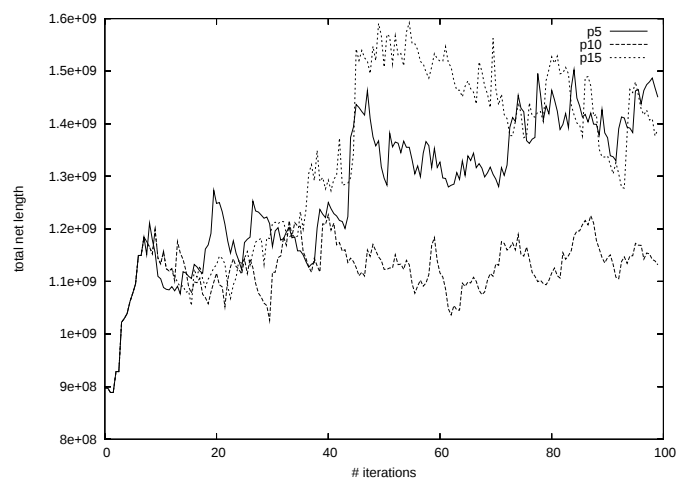


図 6.14: primary1 に対する ML 法による実験結果 (統合 (2)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 3 回, p15 は禁止制約 15 回の結果)

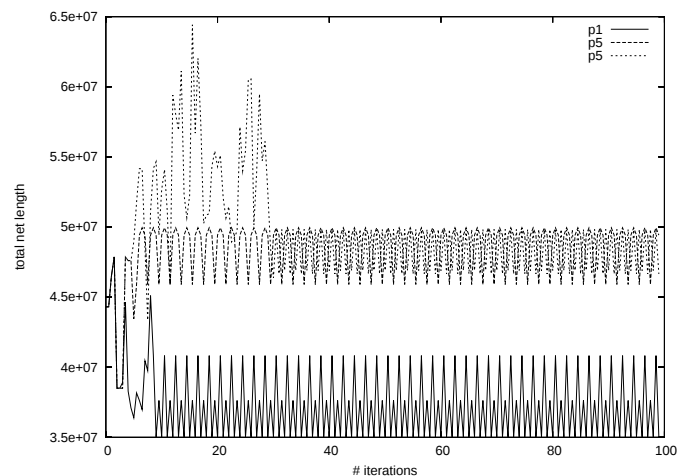


図 6.15: fract に対する MR 法による実験結果 (統合 (2)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p1 は禁止制約 1 回, p3 は禁止制約 3 回, p5 は禁止制約 5 回の結果)

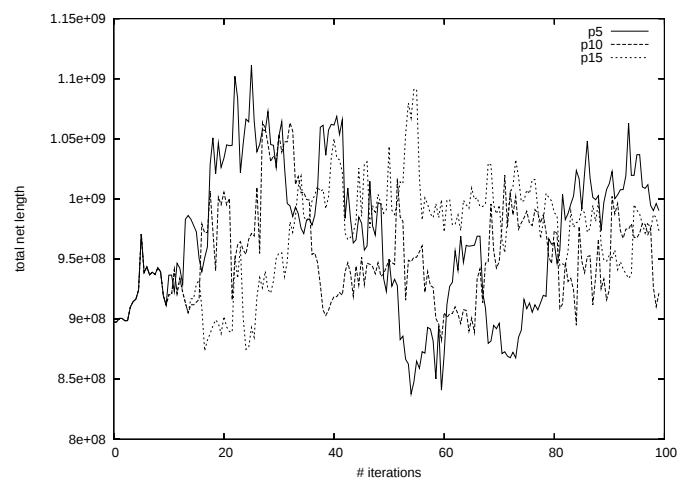


図 6.16: primary1 に対する MR 法による実験結果 (統合 (2)/分割 (2) の繰り返しに対するブロック間配線長の変化, 但し, p5 は禁止制約 5 回, p10 は禁止制約 3 回, p15 は禁止制約 15 回の結果)

表 6.7: fract に対する実験結果 (統合 (2)/分割 (2))

	統合・分割禁止制約		
	1 回	3 回	5 回
<i>ML 法</i>	3.820×10^7	3.945×10^7	3.945×10^7
<i>ML 法</i>	3.502×10^7	3.850×10^7	3.850×10^7

表 6.8: primary1 に対する実験結果 (統合 (2)/分割 (2))

	統合・分割禁止制約		
	5 回	10 回	15 回
<i>ML 法</i>	8.889×10^8	8.889×10^8	8.889×10^8
<i>ML 法</i>	8.373×10^8	8.822×10^8	8.735×10^8

表 6.9: ML 法を用いた統合/分割ブロック選択法における結果比較 (fract)

	統合・分割禁止制約		
	5 回	10 回	15 回
統合 (1)/分割 (1)	4.150×10^7	3.999×10^7	4.002×10^7
統合 (1)/分割 (2)	4.098×10^7	4.142×10^7	4.142×10^7
統合 (2)/分割 (1)	4.151×10^7	4.113×10^7	4.113×10^7
統合 (2)/分割 (2)	3.820×10^7	3.945×10^7	3.945×10^7

表 6.10: ML 法を用いた統合/分割ブロック選択法における結果比較 (primary1)

	統合・分割禁止制約		
	5 回	10 回	15 回
統合 (1)/分割 (1)	8.889×10^8	8.889×10^8	8.889×10^8
統合 (1)/分割 (2)	8.889×10^8	8.889×10^8	8.889×10^8
統合 (2)/分割 (1)	8.889×10^8	8.889×10^8	8.889×10^8
統合 (2)/分割 (2)	8.889×10^8	8.889×10^8	8.889×10^8

表 6.11: MR 法を用いた統合/分割ブロック選択法における結果比較 (fract)

	統合・分割禁止制約		
	5 回	10 回	15 回
統合 (1)/分割 (1)	3.672×10^7	3.755×10^7	4.003×10^7
統合 (1)/分割 (2)	3.599×10^7	3.502×10^7	3.851×10^7
統合 (2)/分割 (1)	4.345×10^7	3.639×10^7	3.444×10^7
統合 (2)/分割 (2)	3.502×10^7	3.850×10^7	3.850×10^7

表 6.12: MR 法を用いた統合/分割ブロック選択法における結果比較 (primary1)

	統合・分割禁止制約		
	5 回	10 回	15 回
統合 (1)/分割 (1)	8.872×10^8	8.707×10^8	7.958×10^8
統合 (1)/分割 (2)	8.975×10^8	8.499×10^8	8.097×10^8
統合 (2)/分割 (1)	8.975×10^8	8.975×10^8	8.889×10^8
統合 (2)/分割 (2)	8.373×10^8	8.823×10^8	8.735×10^8

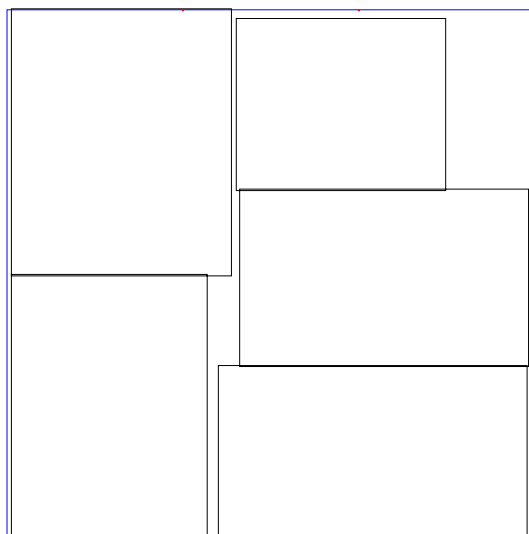


図 6.17: 初期配置 (fract)

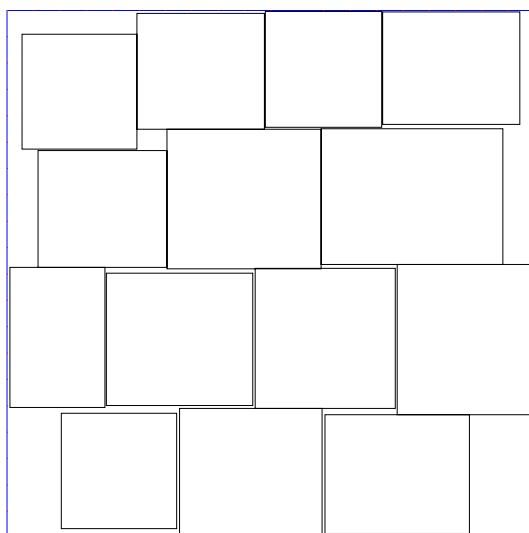


図 6.18: 初期配置 (primary1)

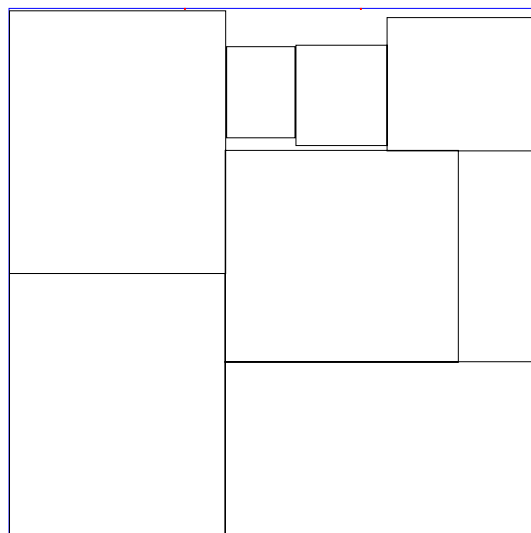


図 6.19: 配線長最小時の配置結果 (fract:禁止制約 3)

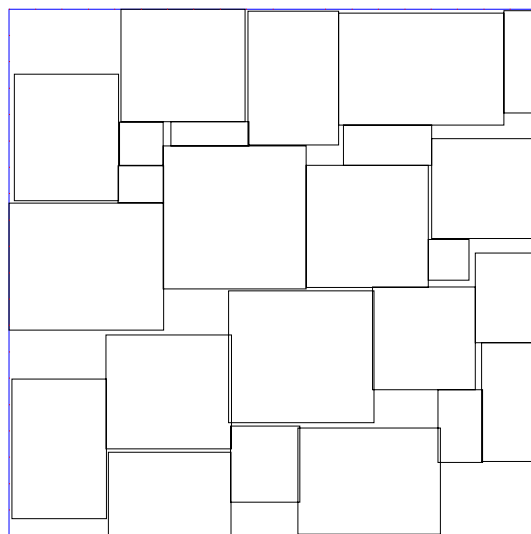


図 6.20: 配線長最小時の配置結果 (primary1:禁止制約 15)

第7章 まとめ

本研究では, FD 法を用いたブロックの概略配置と, そこで用いられる力を指標としたブロックの分割, 統合を繰り返し行うことで, 入力回路の階層記述に捕らわれることなく, レイアウトに適した回路分割とそれに対するフロアプランを得る一手法を提案し, 計算機実験より提案手法が入力として与えられたグラフ分割でのブロック配置に対して自乗配線長を 9.9–21.0 % 改善することを確認した.

今後の課題として, 統合分割に関しては, より適切な統合/分割対象ブロックの選択や実現の手法の提案と, 3 個以上のブロックの統合/分割のアルゴリズムについても考案する. 信号伝播遅延や消費電力などのレイアウトに依存して決まる回路性能を重視した回路分割手法の検討があげられる.

謝辞

本研究を進めるにあたり、終始適切なご助言と暖かいご指導をいただいた北陸先端科学技術大学院大学 金子 峰雄教授、同 田湯 智助手、同 高島 康裕助手、そして研究室の学生のみなさまに深く感謝致します。また、勉学の機会を与えてくれた両親と家族に感謝致します。

参考文献

- [1] Jason Cong. *An Interconnect-Centric Design Flow for Nanometer Technologies*, SASIMI2001, pp.199–205, 2001.
- [2] Neil R. Quinn, Jr. and Melvin A Breuer. *A Forced Directed Component Placement Procedure for Printed Circuit Boards*, IEEE Trans. on Circuits and Systems, Vol. CAS, No. 26, pp.377–388, 1979.
- [3] Hans Eisenmann and Frank M. Johannes. *Generic Global Placement and Floorplanning*, 35th DAC, pp.269–274, 1998.
- [4] 横丸 敏彦, 金澤 裕治, 澁谷 利行. *Soft Block Placement method based on Force Directed Algorithm*, IEICE, VLD2000-137, ICD2000-213 pp.19–24, 2001.