JAIST Repository
https://dspace.jaist.ac.jp/

Title	ソフトモジュールを含む配置問題の一解法
Author(s)	三輪, 剛史
Citation	
Issue Date	1997-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1061
Rights	
Description	Supervisor:平石 邦彦, 情報科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

ソフトモジュールを含む配置問題の一解法

指導教官 平石 邦彦 助教授

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

三輪 剛史

1997 年 2 月 14 日

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY,
HOKURIKU 1990

Copyright © 1997 by Takeshi MIWA

目 次

1 緒論 **3**

2 VLSI 設計の設計工程及びフロアプラン **5**

2.1 VLSI レイアウト設計 5

2.2 フロアプラン領域最小化問題 9

2.3 定式化 9

2.4 水平及び垂直制約グラフ 10

2.5 従来研究と問題点 10

3 ソフトモジュールサイズ決定問題 **13**

3.1 問題設定 13

3.2 候補パス 13

3.3 解法 15

3.3.1 $X > \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合 15

3.3.2 $X = \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合 18

3.3.3 $X > \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合 18

3.3.4 $X = \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合 18

3.4 2 個のソフトモジュールサイズの決定 19

3.4.1 $X > \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合 19

3.4.2 $X = \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合 21

3.4.3 $X > \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合 22

3.4.4 $X = \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合 22

4 実装と評価 **24**

4.1 ソフトモジュールサイズ決定手法の実装 24

4.2 実験内容 24
4.3 実験結果 26
4.4 実験結果の評価 27

5 結論 **35**

謝辞 **36**

参考文献 **37**

付録 **39**

第 1 章

緒論

VLSI(Very Large Scale Integrated circuit) チップの集積度 (単位面積当りの素子数) は年々増大し，現在では 100 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 1IC(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．それに伴い，VLSI の機能はより一層複雑化し，さらに小量多品種，設計・製造期間の短縮が重要な ASIC(Application Specific Integrated Circuit) と呼ばれる VLSI チップが出現し，その設計は工数・期間ともに人手能力の限界を越えたと言える．

このために電子計算機を用いて VLSI チップの設計を支援，または自動化する，CAD(Computer Aided Design)，また DA(Design Automation) が提案され，研究されてきた．その歴史は古く，LSI の集積度は未だ低かったにも関わらず，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，または DA 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 LSI 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配置配線アルゴリズムの研究が活発に行われている．

配置問題は，配置すべき矩形が色々な大きさである為に困難な問題であり，その中でも，縦横比が可変であるソフトモジュールを取り扱う事はより困難な問題である．ソフト

モジュールを含む配置問題において，与えられた各モジュールの相対的位置関係の下で，チップ面積を最小にする各ソフトモジュールの形状を決定する問題は，フロアプラン領域最小化問題と呼ばれる．フロアプラン領域最小化問題は，全てのモジュールがソフトモジュールである場合，ソフトモジュールと形状が固定されたモジュールが混在する場合の二通りに分けられる．さらにソフトモジュールの形状の自由度関し，各ソフトモジュール毎に用意された形状候補のみが採用できる場合と，任意の幅と高さを採用できる場合がある．

ソフトモジュールに関しては様々な従来研究が行われてきたが，そこで取り扱うソフトモジュールは，各モジュール毎に指定された面積が与えられているものであり，ソフトモジュールを一つの矩形として扱っていた．しかし，コントロール部などのランダムロジックでは，必要とされる面積を満たせば，数個に分割しても構わない場合がある．

そこで，本論文では，複数個の合計面積一定のソフトモジュールを含む配置問題において，与えられた相対的位置関係の下でチップ面積を最小化する各ソフトモジュールの縦横長を求める解析的手法を提案する．

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 設計工程と配置問題の位置付けを述べる．第 3 章では，本研究で提案するソフトモジュールサイズ決定手法について述べる．第 4では，本手法を BSG を用いた最適なフラアプラン探索手法に取り込んだシステムを実装し，実験によりその効果を確かめる．

第 2 章

VLSI 設計の設計工程及びフロアプラン

2.1 VLSI レイアウト設計

VLSI のチップ設計工程を図 2.1に示す．まず，設計工程は，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．即ち VLSI 設計の仕様を満たすアーキテクチャの設計である．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，即ち，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，論理シミュレーションを実行する事によって，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，回路シミュレーターによって回路レベルで所望の論理が実現できていいるかどうかをチェックする．

レイアウト設計では，論理図と部品の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尚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．

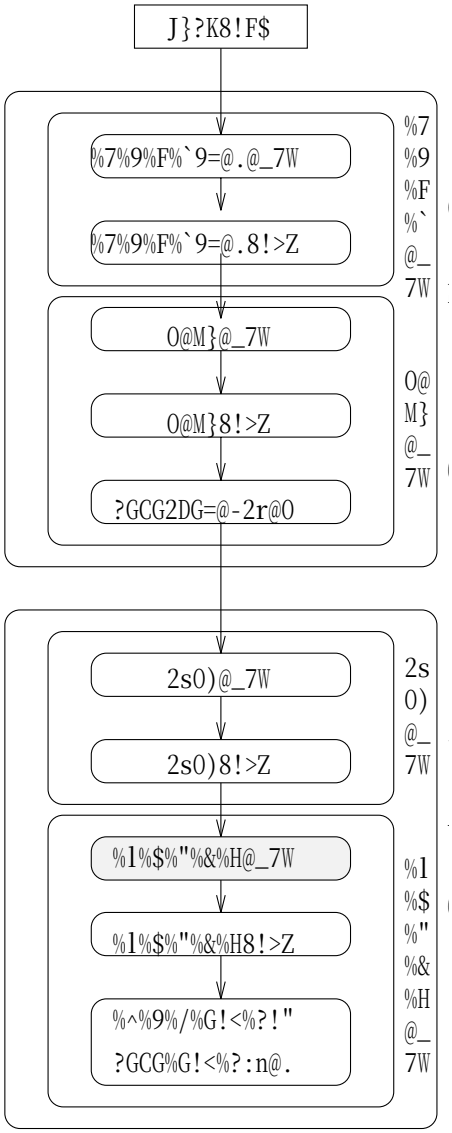


図 2.1: レイアウト設計工程

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方式を説明する．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 3 階層-チップ〜ブロック〜セル-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 (図 2.2参照)．図 2.2に示すのは，ビルディングブロック方式によるチップモデルである．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 (次章からモジュールとする) のレイアウトでは，セルをブラックボックスとみなし，セルを配置してセル間を配線する．チップレベルのレイアウトでは，ブロックを配置してブロック間を配線する．これらの他にチップを配置してチップ間を配線するプリント基板，等のレイアウト設計がある．

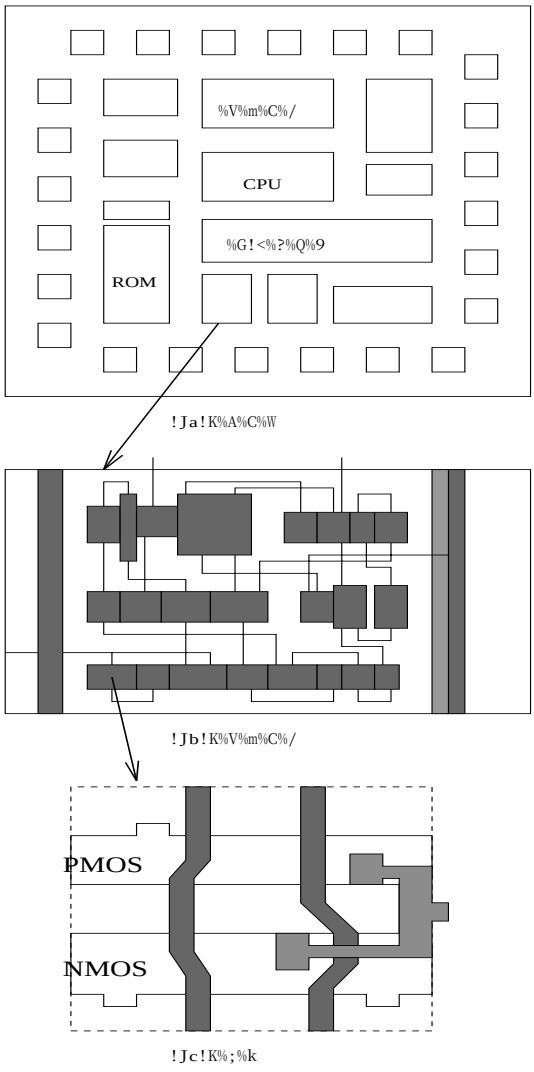


図 2.2: VLSI チップの階層設計方式

これらの配置配線問題を解くための自動レイアウトフローを図 2.3に示す．配置処理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，高精度化が課題であるが，フロアプランについては，未だ手動で行われる事が多く，有効な自動化が待たれている．

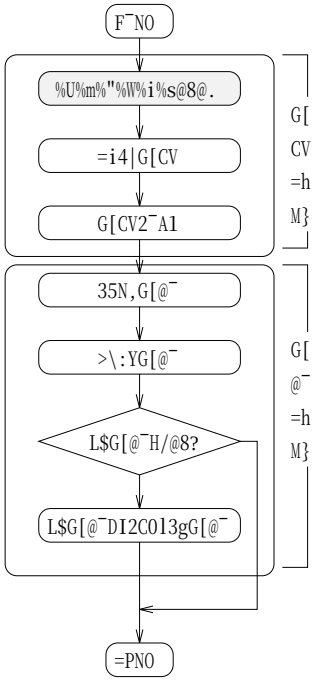


図 2.3: 自動レイアウト処理フロー

階層設計方式

VLSI チップの設計では階層設計方式を取り入れている．その利点は大規模な設計対象を使用メモリ量，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，及び同一階層内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物である．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，セレクタ，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．レイアウト DA プログラムはこれらの各階層毎に，即ちブロック配置，ブロック間配線，ブロック内セル配置配線，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．本論文で対象とするのは，チップ内のブロック配置問題である．

2.2 フロアプラン領域最小化問題

ソフトモジュールを含む配置問題において，相対的位置関係が与えられ，面積を最小にする各モジュールの形状を決定する問題は，フロアプラン領域最小化問題と呼ばれる．フロアプラン領域最小化問題においては，全てのモジュールがソフトモジュールである場合，幾つかがソフトモジュールである場合の二通りに分けられる．さらにソフトモジュールの形状において，各ソフトモジュールにおいて与えられた候補の中から選択する手法と，与えられた各面積ソフトモジュールになるように，任意の幅と高さを求める手法がある．本論文では，与えられたモジュールの幾つかが合計面積一定のソフトモジュールである場合について考える．

2.3 定式化

m 個のモジュールの内，n 個は面積移動可能なモジュール $m_1, m_2 \cdots m_n$ であり，その合計面積 $\mathcal{S}$のみ与えられいる．それ以外のモジュール $m_{n+1}, m_{n+2}, \cdots, m_m$は高さ幅とも与えられているとする．また，$m_1, m_2 \cdots m_m$の相対的位置が与えられている．その時にチップ面積 $X \times Y$，ここで X，Yはチップの幅と高さである，を最小化するように合計面積一定のモジュール $m_1, m_2 \cdots m_n$の高さと幅 (w_1,h_1)，(w_2,h_2)，$\cdots$，(w_n,h_n) を求める問題とする．

$M = m_1, m_2, \cdots, m_n, m_{n+1}, \cdots, m_m$:配置すべき対象の集合
$L = l_1, l_2, \cdots, l_t$:配置可能位置の集合 $(m \leq t)$
$W = w_1, w_2, \cdots, w_n$, $H = h_1, h_2, \cdots, h_n$:ソフトモジュール形状
$s : M \rightarrow L$:一対一写像，配置と呼ぶ.
$X \rightarrow N$, $c(X)$:N は実数の集合
ソフトモジュールの形状を与えた時のチップの幅を示す.
$Y \rightarrow M$, $c(Y)$:M は実数の集合
ソフトモジュールの形状を与えた時のチップの高さを示す.

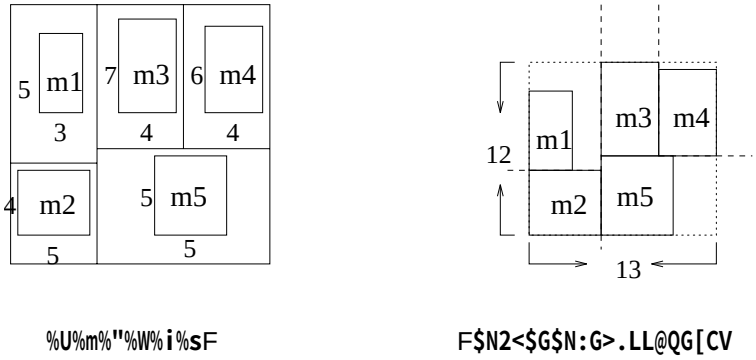
この時，フロアプラン領域最小化問題は次のように定式化できる．
ある配置 $s : M \rightarrow L$ において，

チップ面積:$A(s) = c(X) \times c(Y)$

を最小にするソフトモジュールの形状 X，Y を求めよ．

2.4 水平及び垂直制約グラフ

フロアプランは，基板上でのモジュールの概略的な配置を決定するレイアウト設計の工程の１つであり，チップ全体をモジュールを埋め込む為の領域 (部屋) に隙間無く分割する (図 2.3)．フロアプランは，矩形を分割線と呼ばれる線分を用いて小領域 (部屋) に分割し，各々の部屋に対して高々一つのモジュールを割り当てたものであり，**モジュールと分割線との相対的位置関係**を表す (図 2.4)．上記の条件を満たすフロアプランは，それは，



%U%m%"%W%i%sF　　F$N2<$G$N:G>.LL@QG[CV

図 2.4: フロアプラン

各々一つの sink と一つ source を持つ非循環有向グラフ G_h と G_v としてとらえる事ができる [4] [1]．その時，基矩形の左辺及び右辺を含む各垂直分割線を点に対応させ，辺の重みをルームに埋め込まれたモジュールの幅とした G_h を水平制約グラフとする．同様に，基矩形の上辺及び下辺を含む各水平分割線を点に対応させ，辺の重みをルームに埋め込まれたモジュールの高さとした G_v を垂直制約グラフとする (図 2.5は図 2.4の水平及び垂直制約グラフ)．この，G_h 及び G_v のそれぞれの最大パス長 $\mathcal{W}^c_{\{\}}$，$\mathcal{H}^c_{\{\}}$ が，チップ幅及び高さとなり，その積 $X \times Y$ がチップ面積となる．本研究では，この垂直及び水平制約グラフをモジュールの相対的位置関係の表現として考える．

2.5 従来研究と問題点

ソフトモジュールを含む問題は様々な研究がなされている．その問題は，全てのモジュールがソフトモジュールである場合，ソフトモジュールと形状が固定されたモジュールが混在する場合の二通りに分けられる．さらに，ソフトモジュールの形状の自由度関し，各ソ

[1] 各矩形 R において，分割線分σからσ'への有向な辺 e_r をもつ，ここでσは R の左辺であり，σ' は R の右辺である．このように，各基矩形と G_h の辺との間には一対一対応がある．G_v においても同様である．

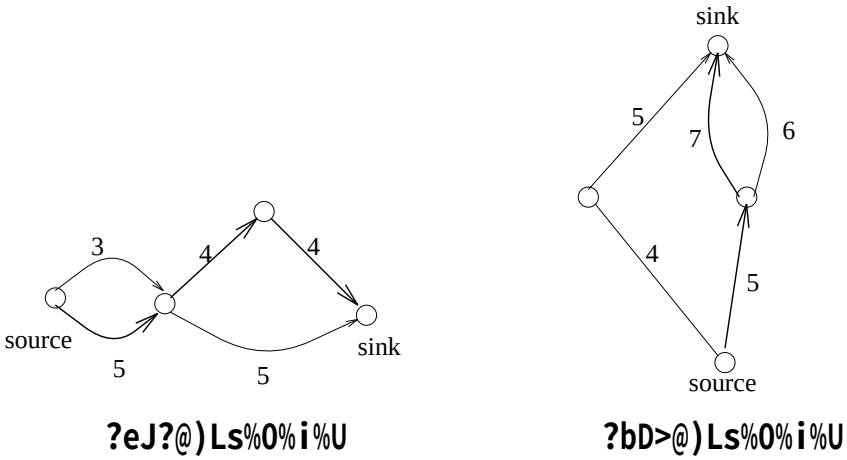


図 2.5: 水平及び垂直制約グラフ

フトモジュール毎に用意された形状候補のみが採用できる場合と，任意の幅と高さを採用できる場合がある．そのいくつかを紹介する．

1. 各ソフトモジュール毎に用意されたの形状のみを採用する問題.

   - slicing structure [4][6][7]

     領域を縦或は横に 2 分割をする事で繰り返しできる構造．Oten & Stockmeyer's アルゴリズムで効率良く最適に特事ができる [5].

   - hierarchical structure [10][11] [12][13][14][15]

     floorplan の中で複数のモジュールからなる矩形領域を sub-floorplan と呼ぶ．sub-floorplan は再帰的に定義され，各 sub-floorplan が高々p 個の sub-floorplan しか含まない時 hierarchical floorplan of p と呼ぶ．sub-floorplan の階層構造は線形オーダー抽出できる事が証明されており [12]．計算量は weakly NP-complete である [15]．また，Oten & Stockmeyer's の手法を拡張する事で，最適解を求める pseudo-polynomial アルゴリズムが提案されている [13].

   - 一般的な構造

     計算量は strongly NP-complete である事が証明されており [6]，[8] では基本的な branch-and-bound 探索アルゴリズムで最適解を求めている．[9] では slicing structure のカットラインの概念を一般の場合に拡張し，最適解を求めている．slicing structure に対しては Otten & Stockmeyer の手法と同じ動作をし，“ ほとんど ”slicing な構造に対しては非常に高速である．

2. ソフトモジュールの形状を任意の幅と高さを採用できる問題

この問題では，無駄な領域ないようにモジュールの形状を決定できるかと言う点に焦点が当てられている．このような無駄な領域の無いフロアプランを zero wasted area floorplan と呼ばれる．最小領域幅制約が無い場合，zero wasted area floorplan は必ず存在し，縦横比の変更したものを同一視すれば一意である事が証明されている [1]．また，最小幅制約がある場合は，はじめに制約無しで求め，それを基に最小幅制約を満たすようにモジュール縦横比を変更する事で対処する手法が提案されている [1][2]．zero wasted area floorplan が可能な場合はこの方法で求められるが，不可能な場合に面積最小解を求める方法については [3] において解けることが示されている．

これらの従来研究で扱われるソフトモジュールは，各ソフトモジュール毎に指定された面積が与えられているものであり，ソフトモジュールは一つの矩形として扱っていた．しかし，コントロール部などのランダムロジックでは，必要とされる面積だけを満たせば数個に分割しても構わない場合がある．これに対応する従来研究はほとんど無く，このようなモジュールを取り扱える要求が出ている．

第 3 章

ソフトモジュールサイズ決定問題

3.1 問題設定

$m+n$ 個のモジュールがあり，その内の m 個については縦横の長さが与えられている．残りの n 個のモジュール $s_1, s_2, \cdots, s_n$は，その合計面積$\mathcal{S}$だけが指定されたソフトモジュールである．すなわち，ソフトモジュール s_iの横幅を w_i，縦幅を h_iとして，与えられた$\mathcal{S}$に対して，

$$\mathcal{S} = \sum_{i=1}^{n} w_i h_i$$

が要請される．

また同時に，フロアプランが与えられ，$m+n$ 個のモジュール全てが各々，異なる部屋に割り当てられている．

以上の条件の下で，チップ全体の面積を最小にする各々のソフトモジュールの縦横の長さを求めよ．

3.2 候補パス

ソフトモジュールの集合 $S(0 \leq |S| \leq n)$ に対して，水平制約グラフ G_h上で，Sに含まれる全てのソフトモジュールに対応する枝を通るパス（Sに含まれないソフトモジュールに対応する枝については，パスに含まれるか否かを問わない）が存在するなら，Sは「水平に有効である」と呼ぶ．垂直についても同様である．

ソフトモジュールに対応する枝の重みを全て 0 にしたグラフ上で，水平に有効なソフ

トモジュール集合 Sに含まれる全てのソフトモジュールに対応する枝を通るパスの中で最長のパス（これを「候補パス」と呼ぶ）の長さを，$\mathcal{W}_S^c$と表記し,「候補パスの基長」と呼ぶ．なお，$\mathcal{W}_S^c$に関して以下の性質が成り立つ．

性質： S_1, S_2は共に水平方向に有効であり，$S_1 \subset S_2$のとき，$\mathcal{W}_{S_1}^c \geq \mathcal{W}_{\{S_2\}}^c$である．

何らかの方法で，全てのソフトモジュールの幅が決められたとして，ある水平に有効なソフトモジュール集合 Sについて，以下の式 (3.1) にて「候補パス長」l_Sを定義する．

$$l_S = \left(\sum_{i=1}^{n} b_i w_i\right) + \mathcal{W}_S^c \tag{3.1}$$

$$\text{但し, } b_i = \begin{cases} 0 & \text{if } s_i \notin S \\ 1 & \text{if } s_i \in S \end{cases}$$

このとき，l_Sは Sに含まれないソフトモジュールに対応する枝の重みを 0 とみなしたときの Sに含まれるソフトモジュールに対応する枝全てを通るパスの中で最長のパスの長さとなる．

水平方向に有効なソフトモジュールの集合全てについて，候補パス（候補パス長）を求めれば，その中の少なくとも一つは，与えられたソフトモジュール幅の下での最長パス（最長パス長）となる．

水平方向に有効なソフトモジュールの集合の全体数は，$n+1$ 以上 2^n以下である．この水平方向の候補パス長全てを行列を用いて表現すると，以下の式の様に書ける．

$$\begin{pmatrix} l_{\{\}} \\ l_{\{s_1\}} \\ l_{\{s_2\}} \\ \vdots \\ l_S \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ b_1 & b_2 & \dots & b_n \\ \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} + \begin{pmatrix} \mathcal{W}_{\{\}}^c \\ \mathcal{W}_{\{s_1\}}^c \\ \mathcal{W}_{\{s_2\}}^c \\ \vdots \\ \mathcal{W}_S^c \\ \vdots \end{pmatrix}$$

$$\text{但し, } b_i = \begin{cases} 0 & \text{if } s_i \notin S \\ 1 & \text{if } s_i \in S \end{cases}$$

候補パスには，以下の定理 1 が成り立つ．

定理1　任意のソフトモジュールサイズ決定問題について，チップ全体の横方向における長さ X が $\mathcal{W}^c_{\{\}}$ よりも大きい最適解の中には，n 以上の本数の水平方向の候補パスの長さが X と等しいものが必ず存在する．n 本の水平方向の候補パスは，後述の式 (3.2) の b^j_i に記した行列記述したときに独立であるとしてよい．垂直方向も同様である．　■

この定理の証明は，n 未満の候補パス長のみが X と等しいと仮定した場合に，矛盾（ソフトモジュールの高さを変化させることなく，より小さいチップ全体の横幅を持つ解が存在する）を導くことにより，なされる (付録参照)．

この定理を基礎として，ソフトモジュールサイズ決定問題の最適解を次節で述べる手法により解析的に求めることができる．

3.3　解法

チップ全体の水平方向における長さを X，垂直方向における長さを Y とする．X と $\mathcal{W}^c_{\{\}}$ の大小関係及び，Y と $\mathcal{H}^c_{\{\}}$ の大小関係により 2×2 通りの場合分けができ，各々の場合について最もチップ面積を小さくするものを求める．ついで，これら４つの中で最もチップ面積を小さくするものを求めれば，これが全体の最適解である．

なお，$X < \mathcal{W}^c_{\{\}}$ や $Y < \mathcal{H}^c_{\{\}}$ は明らかに成り立たない．よって，水平方向について $X = \mathcal{W}^c_{\{\}}$ の場合と $X > \mathcal{W}^c_{\{\}}$ の場合に分け，同様に垂直方向について $Y = \mathcal{H}^c_{\{\}}$ の場合と $Y > \mathcal{H}^c_{\{\}}$ の場合に分ける．

3.3.1　$X > \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合

水平及び垂直両方向について，独立な候補パス n 本の組合せ毎にそれらが最長値（X もしくは Y）と等しい値になるものとして，そのときに最もチップ面積を小さくするもの $(XY \rightarrow min)$ を求める．これら候補パスの組合せの中に，$X > \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合において最もチップ面積を小さくするものが含まれていることは，定理 1 から明らかである．

以下，各ソフトモジュールの幅 w_i と高さ h_i，チップの横長 X，縦長 Y をそれぞれ変数と考える．

いま，水平方向について，全ての候補パスの中からある n 個の独立な候補パスを取り出し，それらの候補パス長がチップ横長 X と等しく，また，垂直方向についても同様に，ある n 個の独立な候補パスを取り出し，それらの候補パス長がチップ縦長 Y と等しいと

した場合のチップ面積最小化について考える．

水平方向の候補パス長に関する制約は，選ばれた n 個の候補パスに関して，

$$\begin{pmatrix} X \\ X \\ \vdots \\ X \end{pmatrix} = \begin{pmatrix} b_1^1 & b_2^1 & \dots & b_n^1 \\ b_1^2 & b_2^2 & \dots & b_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ b_1 & b_2 & \dots & b_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} + \begin{pmatrix} \mathcal{W}^c_{S^1} \\ \mathcal{W}^c_{S^2} \\ \vdots \\ \mathcal{W}^c_{S^n} \end{pmatrix} \tag{3.2}$$

$$\text{但し, } b_i^j = \begin{cases} 0 & \text{if } s_i \notin S^j \\ 1 & \text{if } s_i \in S^j \end{cases}$$

一方，選ばれなかった残りの候補パス全てに関して，

$$X \geq \sum_{i=1}^{n} b_i^{n+k} w_i + \mathcal{W}^c_{S^{n+k}}$$

と記述することができる．

前者において，係数行列が正則であることから，逆行列が存在し，各ソフトモジュールの幅 w_i について解くことができる．

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} b_1^1 & b_2^1 & \dots & b_n^1 \\ b_1^2 & b_2^2 & \dots & b_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ b_1 & b_2 & \dots & b_n \end{pmatrix}^{-1} \left(\begin{pmatrix} X \\ X \\ \vdots \\ X \end{pmatrix} - \begin{pmatrix} \mathcal{W}^c_{S^1} \\ \mathcal{W}^c_{S^2} \\ \vdots \\ \mathcal{W}^c_{S^n} \end{pmatrix} \right)$$

これにより各 w_iは 変数 Xについての 1 次式として，

$$w_i = \alpha_i X + \beta_i \tag{3.3}$$

の形にて表すことができる（α_i, β_iは定数）．ここで，$w_i \geq 0$ が必要であることから，X に関する以下の制約を得ることができる．

$$\begin{aligned} X &\geq -\tfrac{\beta_i}{\alpha_i} \quad \text{if } \alpha_i > 0 \\ X &\leq -\tfrac{\beta_i}{\alpha_i} \quad \text{if } \alpha_i < 0 \end{aligned}$$

一方，これら w_iを候補パスについての後者の制約に代入すれば，

$$X \geq \sum_{i=1}^{n} b_i^{n+k} (\alpha_i X + \beta_i) + \mathcal{W}^c_{S^{n+k}}$$

すなわち，

$$X \geq \frac{\sum_{i=1}^{n} b_i^{n+k}\beta_i + \mathcal{W}_S^{c\,n+k}}{1-\sum_{i=1}^{n} b_i^{n+k}\alpha_i} \quad : \text{when } 1-\sum_{i=1}^{n} b_i^{n+k}\alpha_i > 0$$
$$X \leq \frac{\sum_{i=1}^{n} b_i^{n+k}\beta_i + \mathcal{W}_S^{c\,n+k}}{1-\sum_{i=1}^{n} b_i^{n+k}\alpha_i} \quad : \text{when } 1-\sum_{i=1}^{n} b_i^{n+k}\alpha_i < 0$$

以上で得られる X に関する制約全てを満足する実数上の範囲が存在しないとき，選ばれた n 個の候補パス長が最大パス長となることはないと判断して，別の n 個の候補パス長の選択へ移る．この範囲が存在するとき，それを $[X_{min}, X_{max}]$ とする．

以上，水平方向の候補パス長，チップの横幅，モジュールの横幅に関して考えたが，垂直方向の候補パス長，チップの縦長，モジュールの縦長に関して同様の議論から，各ソフトモジュールの縦長をチップの縦長変数 Y を使って，

$$h_j = \gamma_j Y + \varepsilon_j \tag{3.4}$$

と表すことができる（γ_j, ε_j は定数）とともに，Y の値の範囲 $[Y_{min}, Y_{max}]$ が得られる．

以上から，ここで解くべき問題は，

$$\begin{aligned} S &= \sum_{i=1}^{n} w_i h_i \\ &= \zeta XY + \eta X + \vartheta Y + \iota \end{aligned} \tag{3.5}$$

但し，

$$\zeta = \textstyle\sum_{i=1}^{n} \alpha_i \gamma_i$$
$$\eta = \textstyle\sum_{i=1}^{n} \alpha_i \varepsilon_i$$
$$\vartheta = \textstyle\sum_{i=1}^{n} \gamma_i \beta_i$$
$$\iota = \textstyle\sum_{i=1}^{n} \beta_i \varepsilon_i$$

を満足し，かつ値の範囲 $X_{min} \leq X \leq X_{max}$, $Y_{min} \leq Y \leq Y_{max}$ の中で，チップ面積 XY を最小にする X と Y を求める問題となる．この問題の解は，(1) 変数値の取り得る範囲の境界，

$$\begin{array}{ll} \left(X_{min}, \frac{S-\eta X_{min}-\iota}{\zeta X_{min}+\vartheta}\right) & \text{when } Y_{min} \leq \frac{S-\eta X_{min}-\iota}{\zeta X_{min}+\vartheta} \leq Y_{max} \\ \left(X_{max}, \frac{S-\eta X_{max}-\iota}{\zeta X_{max}+\vartheta}\right) & \text{when } Y_{min} \leq \frac{S-\eta X_{max}-\iota}{\zeta X_{max}+\vartheta} \leq Y_{max} \\ \left(\frac{S-\vartheta Y_{min}-\iota}{\zeta Y_{min}+\eta}, Y_{min}\right) & \text{where } X_{min} \leq \frac{S-\vartheta Y_{min}-\iota}{\zeta Y_{min}+\eta} \leq X_{max} \\ \left(\frac{S-\vartheta Y_{max}-\iota}{\zeta Y_{max}+\eta}, Y_{max}\right) & \text{where } X_{min} \leq \frac{S-\vartheta Y_{max}-\iota}{\zeta Y_{max}+\eta} \leq X_{max} \end{array}$$

あるいは (2) 制約式を目的関数に代入して得られる関数

$$\begin{aligned} f(X) &= XY \\ &= X \cdot \frac{S-\eta X-\iota}{\zeta X+\vartheta} \end{aligned} \tag{3.6}$$

の X に関する極小点にて与えられることから，これらの点の中で値の範囲に入る点について，実際に値 XY を評価し，その中で最小を与える X，Y が解となる．

3.3.2 $X = \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合

この場合，独立な横方向の候補パス n 本が $X = \mathcal{W}^c_{\{\}}$ と等しいとして，3.3.1 の場合と同様に，最もチップ面積を小さくするもの（この場合では縦方向の長さを最小とするもの）を求めればよい．

この方法の正当性は，もし横方向の候補パスの中，n 未満の候補パスの候補パス長が $X = \mathcal{W}^c_{\{\}}$ に等しいとした場合に，チップの横幅と各ソフトモジュールの縦長を変えずに，ソフトモジュールの合計面積をより大きく出来ることに因る．

3.3.3 $X > \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合

3.3.2 の場合と同様にして解くことができる．

3.3.4 $X = \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合

この時，次の補助問題を考える．

$X = \mathcal{W}^c_{\{\}}, Y = \mathcal{H}^c_{\{\}}$ の下で，

$$\sum_i^n h_i w_i$$

を最大にせよ．

この問題の解 h_i^*, w_i^* は，もともとの問題で指定されたソフトモジュールの合計面積 S に対して，

$$\sum_i^n h_i^* w_i^* \geq S$$

が成り立つことは明らかである．従って，この補助問題の解を使い，幾つかの方法で，もともとの問題の最適解を導ける．例えば，補助問題の解に対して，

$$h_i = h_i^* \tag{3.7}$$

$$w_i = w_i^* \times \frac{S}{\sum_i^n h_i^* w_i^*} \tag{3.8}$$

にて，所望の解を得る．

ここで導入した補助問題に関して，定理１と同様に，独立な横方向の候補パス n 本が $X = \mathcal{W}^c_{\{\}}$ と等しく，独立な縦方向の候補パス n 本が $Y = \mathcal{H}^c_{\{\}}$ と等しい最適解が存在することが証明できることから，3.3.1 と同様にして，候補パスの組を替えながら最適解を探索していくことが出来る．なおこのとき，補助問題の真の最適解は必要なく，

$$\sum_i^n h_i w_i \geq S$$

を満足する解が得られた時点で探索を打ち切っても，もともとの問題の解が得られる．

3.4　2 個のソフトモジュールサイズの決定

特に 2 個の場合について具体的に示す．この場合，一般性を失う事なく $\{s_1\}$ と $\{s_2\}$ は水平方向には有効であり，垂直方向には，有効でないとする．

$$\begin{pmatrix} l_{\{\}} \\ l_{\{s_1\}} \\ l_{\{s_2\}} \\ l_{\{s_1,s_2\}} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \begin{pmatrix} \mathcal{W}^c_{\{\}} \\ \mathcal{W}^c_{\{s_1\}} \\ \mathcal{W}^c_{\{s_2\}} \\ \mathcal{W}^c_{\{s_1,s_2\}} \end{pmatrix} \tag{3.9}$$

$$\begin{pmatrix} m_{\{\}} \\ m_{\{s_1\}} \\ m_{\{s_2\}} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + \begin{pmatrix} \mathcal{H}^c_{\{\}} \\ \mathcal{H}^c_{\{s_1\}} \\ \mathcal{H}^c_{\{s_2\}} \end{pmatrix}$$

3.4.1　$X > \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合

水平及び垂直方向において，独立な候補パス 2 本ずつを取り出す組合せは 3 つあり，下記の通りである．

1. $$\begin{cases} X = w_1 + \mathcal{W}^c_{\{s_1\}} = w_2 + \mathcal{W}^c_{\{s_2\}} \\ \qquad\qquad \geq w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$$

$$w_1 = X - \mathcal{W}^c_{\{s_1\}} \qquad w_2 = X - \mathcal{W}^c_{\{s_2\}}$$
$$h_1 = Y - \mathcal{H}^c_{\{s_1\}} \qquad h_2 = Y - \mathcal{H}^c_{\{s_2\}}$$

であり，この時 X, Y の定義域は

$$\begin{aligned} \mathcal{W}^c_{\{\}} &< X \leq \mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_2\}} - \mathcal{W}^c_{\{s_1,s_2\}} \\ \mathcal{H}^c_{\{\}} &< Y \end{aligned}$$

である．Y を X によって表す事ができ，

$$Y = \frac{\mathcal{S} - (\mathcal{W}^c_{\{s_1\}}\mathcal{H}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_2\}}\mathcal{H}^c_{\{s_2\}}) + (\mathcal{H}^c_{\{s_1\}} + \mathcal{H}^c_{\{s_2\}})X}{2X - \mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_2\}}}$$

である．これからチップ面積 $f(X)$ を X の式で表す事ができる．

$$\begin{aligned} f(X) &= X \times Y \\ &= \frac{(\mathcal{H}^c_{\{s_1\}} + \mathcal{H}^c_{\{s_2\}})X^2 + (\mathcal{S} - \mathcal{W}^c_{\{s_1\}}\mathcal{H}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_2\}}\mathcal{H}^c_{\{s_2\}})X}{2X - \mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_2\}}} \end{aligned}$$

X, Y の定義域より $f(X)$ における X の定義域は，

$$\mathcal{W}^c_{\{\}} < X < \frac{\mathcal{S} - (\mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_2\}})(\mathcal{H}^c_{\{s_1\}} - \mathcal{H}^c_{\{s_2\}})}{\mathcal{H}^c_{\{s_1\}} - \mathcal{H}^c_{\{s_2\}}} + \mathcal{W}^c_{\{\}}$$

となる．ここで

$f(X)$ を微分しまとめると．

$$\begin{aligned} &(2X - (\mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_2\}}))^2(\mathcal{H}^c_{\{s_1\}} + \mathcal{H}^c_{\{s_2\}}) - \\ &(2\mathcal{S} + (\mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_2\}})(\mathcal{H}^c_{\{s_1\}} - \mathcal{H}^c_{\{s_2\}}))(\mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_2\}}) = 0 \end{aligned}$$

となり $f(X)$ の極小点 X を求める事ができる．これより，定義域から，実際に XY を評価しその中で最小を与える値を解とする．

2. $$\begin{cases} X = w_1 + \mathcal{W}^c_{\{s_1\}} = w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ \qquad\qquad\qquad\qquad \geq w_2 + \mathcal{W}^c_{\{s_2\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$$

$$\begin{aligned} w_1 &= X - \mathcal{W}^c_{\{s_1\}} \qquad w_2 = X - w_1 - \mathcal{W}^c_{\{s_1,s_2\}} \\ h_1 &= Y - \mathcal{H}^c_{\{s_1\}} \qquad h_2 = Y - \mathcal{H}^c_{\{s_2\}} \end{aligned}$$

であり，この時 X と Y の定義域は

$$\begin{aligned} \mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_1,s_2\}} &\leq X \\ \mathcal{H}^c_{\{\}} &< Y \end{aligned}$$

である．Y を X によって表す事ができる．

$$Y = \frac{\mathcal{S} + (\mathcal{W}^c_{\{s_1\}}\mathcal{H}^c_{\{s_2\}} - \mathcal{W}^c_{\{s_1\}}\mathcal{H}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_1,s_2\}}\mathcal{H}^c_{\{s_2\}}) + \mathcal{H}^c_{\{s_1\}}X}{X - \mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_1,s_2\}}}$$

これよりチップ面積 $f(X)$ を X の式で表す事ができる.

$$f(X) = = \frac{\mathcal{H}^c_{\{s_1\}}X^2 + (\mathcal{S} + (\mathcal{W}^c_{\{s_1\}}\mathcal{H}^c_{\{s_2\}} - \mathcal{W}^c_{\{s_1\}}\mathcal{H}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_1,s_2\}}\mathcal{H}^c_{\{s_2\}}))X}{X - \mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_1,s_2\}}}$$

これより，1. と同様にこの式の極小点を求め，定義域から，実際に XY を評価しその中で最小を与える値を解とする．

3. $\begin{cases} X = w_2 + \mathcal{W}^c_{\{s_2\}} = w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ \qquad\qquad\qquad\qquad \geq w_1 + \mathcal{W}^c_{\{s_1\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$

一般性を失わず，2. と同様に解く事ができる．

3.4.2 $X = \mathcal{W}^c_{\{\}}$ かつ $Y > \mathcal{H}^c_{\{\}}$ の場合

計算方法は 3.4.1 と同様であるので，場合分けと w_1, w_2，h_1, h_2 をそれぞれ X,Y で表した式と X，Y 定義域のみ示す.

1. $\begin{cases} X = \mathcal{W}^c_{\{\}} = w_1 + \mathcal{W}^c_{\{s_1\}} = w_2 + \mathcal{W}^c_{\{s_2\}} \\ \qquad\qquad\qquad\qquad \geq w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$

$$\begin{aligned} w_1 &= X - \mathcal{W}^c_{\{s_1\}} \qquad w_2 = X - \mathcal{W}^c_{\{s_2\}} \\ h_1 &= Y - \mathcal{H}^c_{\{s_1\}} \qquad h_2 = Y - \mathcal{H}^c_{\{s_2\}} \end{aligned}$$

$$\begin{aligned} \text{定義域}: \qquad & X = \mathcal{W}^c_{\{\}} \leq \mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_2\}} - \mathcal{W}^c_{\{s_1,s_2\}} \\ & Y > \mathcal{H}^c_{\{\}} \end{aligned}$$

2. $$\begin{cases} X = \mathcal{W}^c_{\{\}} = w_1 + \mathcal{W}^c_{\{s_1\}} = w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \geq w_2 + \mathcal{W}^c_{\{s_2\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$$

$$\begin{aligned} w_1 &= X - \mathcal{W}^c_{\{s_1\}} \qquad & w_2 &= X - w_1 - \mathcal{W}^c_{\{s_1,s_2\}} \\ h_1 &= Y - \mathcal{H}^c_{\{s_1\}} \qquad & h_2 &= Y - \mathcal{H}^c_{\{s_2\}} \end{aligned}$$

$$\begin{aligned} \text{定義域}: \qquad & \mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_1\}} - \mathcal{W}^c_{\{s_1,s_2\}} \leq X = \mathcal{W}^c_{\{\}} \\ & Y > \mathcal{H}^c_{\{\}} \end{aligned}$$

3. $$\begin{cases} X = \mathcal{W}^c_{\{\}} = w_2 + \mathcal{W}^c_{\{s_2\}} = w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \geq w_1 + \mathcal{W}^c_{\{s_1\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$$

一般性を失わず，2 と同様に解く事ができる．

3.4.3 $X > \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合

一般性を失わず 3.4.2 と同様に解く事ができる．

3.4.4 $X = \mathcal{W}^c_{\{\}}$ かつ $Y = \mathcal{H}^c_{\{\}}$ の場合

水平及び垂直方向において，独立な候補パス 2 本ずつを取り出す組合せは 3 つある．

1. $$\begin{cases} X = \mathcal{W}^c_{\{\}} = w_1 + \mathcal{W}^c_{\{s_1\}} = w_2 + \mathcal{W}^c_{\{s_2\}} \\ \qquad\qquad\qquad\qquad \geq w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$$

水平方向に $w_1, w_2,$ 垂直方向において h_1, h_2 をそれぞれ X, Y の一次式及び X，Y の定義を以下に示す．

$$\begin{aligned} w_1 &= X - \mathcal{W}^c_{\{s_1\}} \qquad & w_2 &= X - \mathcal{W}^c_{\{s_2\}} \\ h_1 &= Y - \mathcal{H}^c_{\{s_1\}} \qquad & h_2 &= Y - \mathcal{H}^c_{\{s_2\}} \end{aligned}$$

$$\text{定義域}: \quad X = \mathcal{W}^c_{\{\}} \leq \mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_2\}} - \mathcal{W}^c_{\{s_1,s_2\}}$$
$$Y = \mathcal{H}^c_{\{\}}$$

これより，w_1, w_2, h_1, h_2の値が求まる．定義域の条件を満たし，$w_1h_1+w_2h_2 \geq \mathcal{S}$ならば，合計面積$\mathcal{S}$となるよう，$w_1, w_2$の道を次の様に調整する．

1)$w_1h_1 > \mathcal{S}$なら，

$$w_1 = \frac{\mathcal{S}}{h_1} \qquad w_2 = 0$$

2)$w_2h_2 > \mathcal{S}$なら，

$$w_2 = \frac{\mathcal{S}}{h_2} \qquad w_1 = 0$$

3) 1)2) 以外なら

$$w_2 = \frac{\mathcal{S} - w_1h_1}{h_2}$$

2.
$$\begin{cases} X = \mathcal{W}^c_{\{\}} = w_1 + \mathcal{W}^c_{\{s_1\}} = w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \geq w_2 + \mathcal{W}^c_{\{s_2\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$$

水平方向に$w_1, w_2,$垂直方向においてh_1, h_2をそれぞれX, Yの一次式及びX，Yの定義を以下に示す.

$$w_1 = X - \mathcal{W}^c_{\{s_1\}} \qquad w_2 = X - w_1 - \mathcal{W}^c_{\{s_1,s_2\}}$$
$$h_1 = Y - \mathcal{H}^c_{\{s_1\}} \qquad h_2 = Y - \mathcal{H}^c_{\{s_2\}}$$

$$\text{定義域}: \quad \mathcal{W}^c_{\{s_1\}} + \mathcal{W}^c_{\{s_2\}} - \mathcal{W}^c_{\{s_1,s_2\}} \leq X = \mathcal{W}^c_{\{\}}$$
$$Y = \mathcal{H}^c_{\{\}}$$

これより，w_1, w_2, h_1, h_2の値が求まる．定義域の条件を満たし，$w_1h_1+w_2h_2 \geq \mathcal{S}$ならば，合計面積$\mathcal{S}$となるように，$w_1, w_2$を 1. と同様に調整する．

3.
$$\begin{cases} X = \mathcal{W}^c_{\{\}} = w_2 + \mathcal{W}^c_{\{s_2\}} = w_1 + w_2 + \mathcal{W}^c_{\{s_1,s_2\}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \geq w_1 + \mathcal{W}^c_{\{s_1\}} \\ Y = h_1 + \mathcal{H}^c_{\{s_1\}} = h_2 + \mathcal{H}^c_{\{s_2\}} \end{cases}$$

一般性を失わずに，2 と同様にできる．

第 4 章

実装と評価

4.1 ソフトモジュールサイズ決定手法の実装

前章までに与えられたモジュールの中に複数個のソフトモジュールが存在する場合，水平及び垂直制約グラフにおける最大パス長の積を最小化するように，各ソフトモジュールサイズを決定する手法について示した．

しかし，実際のレイアウト配置では，様々なフロアプランの中から良いものを探索しなくてはならない．そこで，様々なフロアプランの中から良いもの探索する手法として，中武ら [17] と同様に，SA を BSG(図 4.1) に適用した手法を使う．BSG はメタグリッドと呼ばれるモジュールの位置や回転方向などの抽象的な情報を保持するデータ構造であり，他に Sequence-Pair[16] の 2 種類が提案されいる．これに本論文で提案したソフトモジュールサイズ決定手法を組み入れる．

実際には，合計面積が指定されるソフトモジュールを 2 つ含む場合についてアルゴリズムの構築し，計算機への実装を行う．

中武ら [17] に習い，評価関数はモジュールを囲む最小矩形の面積とし，アニーリングスケジュールは標準的なものを使用した．

尚，プログラム，実験は Sun の Sparc Station 5 上で行った．

4.2 実験内容

本実験での対象データを示す．本実験で用いたデータはランダムに矩形を 30 個作成したものと，VLSI レイアウト CAD 用の MCNC ベンチマーク ami33 である．本実験では，

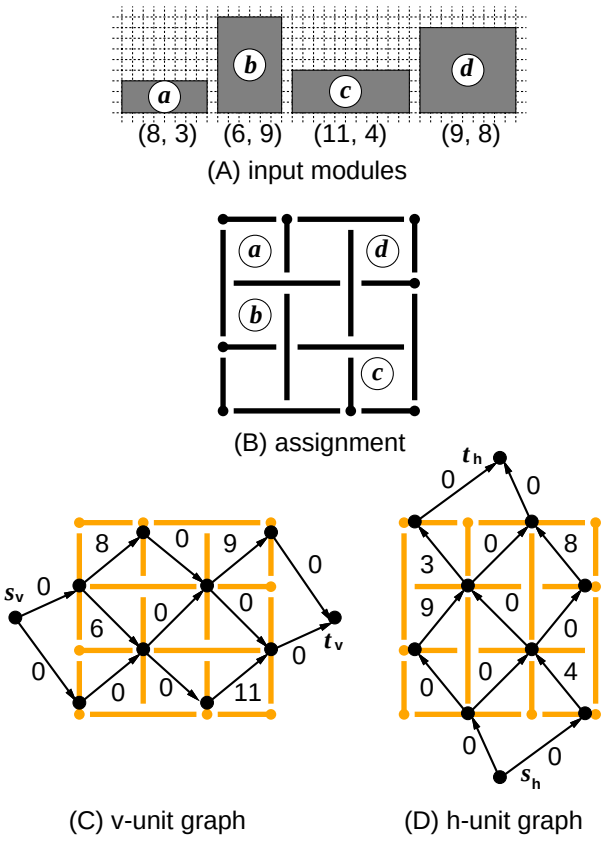


図 4.1: BSG

ソフトモジュール形状最適化手法の効果を確かめる為，下記の 3 つの場合について，面積による評価をする．

1. 全てが形状固定のモジュール．

2. 1 個のソフトモジュールのサイズを最適化したもの．

3. 2 個の面積合計一定ソフトモジュールのサイズを最適化したもの．

ami33 の全てのモジュールは縦横の長さが規定されているが，その中で最も大きな面積のものをソフトモジュールとして扱う．また，合計面積一定モジュールを 2 個については，1 つの縦横の長さが決まったモジュールを 2 つに分割して作る事にする[1]．

尚，実験では，SA の 1 温度における繰り返し回数[2]毎それぞれで，使用するランダムの種を変え 10 回ずつ試行する．

[1] この場合取り扱うモジュール数は 34 個となる

[2] 各温度毎に隣接解を生成する回数

表 4.1: 本研究で実験に用いたデータ

	ランダム作成	ami33
モジュール数	30	33
モジュールの合計面積	469	1156449
ソフトモジュールの合計面積	100	74480

4.3 実験結果

繰り返し数の変化に対する，チップ面積の平均値，最良値，最悪値をそれぞれ図4.2,4.3,4.4，図4.9,4.10,4.11に示す．図4.5，4.12は，固定モジュールの，平均値，最良値，最悪値の三つを示したものであり，図4.6，図4.13は，面積合計一定モジュールが2個の場合の，平均値，最良値，最悪値を示したものである．それぞれ，縦軸がチップ面積を表しており，横軸が繰り返し数である．また，ランダムに作成した30個のモジュールで繰り返し数が100,200の場合の具体的な値をそれぞれ表4.2，ami33の繰り返し数が150,300の時の具体的な値を表4.3に示す．それぞれの平均処理時間を表4.3，表4.3示す．そしてソフトモジュール0個の場合と2個の場合の実際の配置を図4.7，図4.8に示す．

表 4.2: ランダムデータ30個での100,200での評価値

	100			200		
	平均値	最良値	最悪値	平均値	最良値	最悪値
0個の場合	521.20	500.00	532.00	518.10	500.00	528.00
1個の場合	512.20	498.33	522.22	503.08	493.33	513.00
2個の場合	498.10	485.36	506.00	495.61	488.54	504.00

表 4.3: ami33での150,300評価値評価値

	150			300		
	平均値	最良値	最悪値	平均値	最良値	最悪値
0個の場合	1280447	1265180	1320550	1263656	1240657	1315748
1個の場合	1278802	1256850	1309476	1257468	1239504	1277099
2個の場合	1262965	1245531	1278459	1233819	1219447	1254792

表 4.4: ランダムデータ 30 個における平均処理時間 (sec)

繰り返し数	100	200
0 個の場合	44.95	89.67
1 個の場合	53.21	125.38
2 個の場合	135.01	298.11

表 4.5: ami33 における平均処理時間 (sec)

繰り返し数	150	300
0 個	53.88	112.22
1 個	70.36	145.07
2 個	176.51	364.32

4.4 実験結果の評価

形状固定モジュールだけを配置する場合よりも，ソフトモジュールを 1 個含む場合の配置領域面積が小さく，さらに，合計面積一定のモジュールを 2 個含む場合，より一層の配置領域面積の縮小を実現できた．一方，計算時間に関しては，2 個のソフトモジュールを含む場合と形状固定モジュールのみ場合を比べると，約 3.3 倍となり，計算時間の短縮が今後の課題である．方がさらに面積が小さくなった．しかし，計算時間においては，2 個の場合は 0 個の時の約 3.3 倍程度になり，これが今後の課題である．

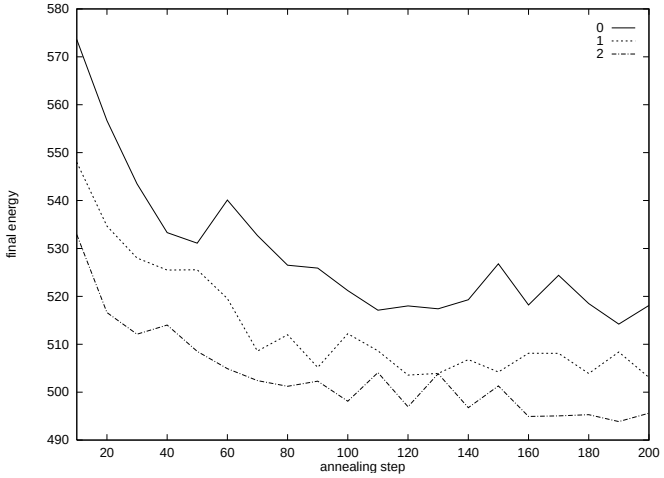


図 4.2: 平均値の推移 (ランダム)

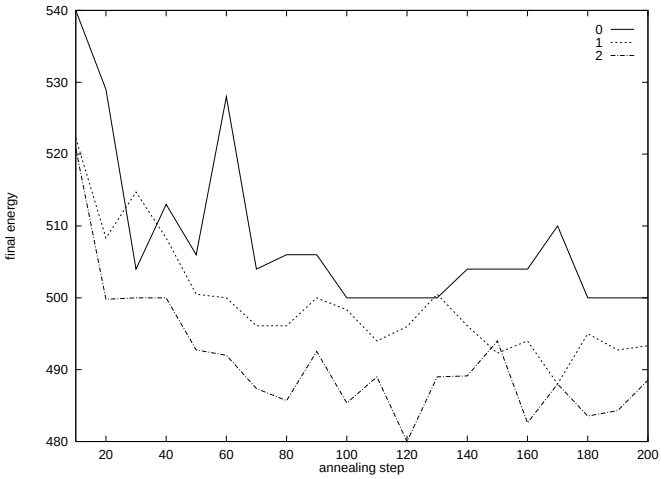


図 4.3: 最良値の推移 (ランダム)

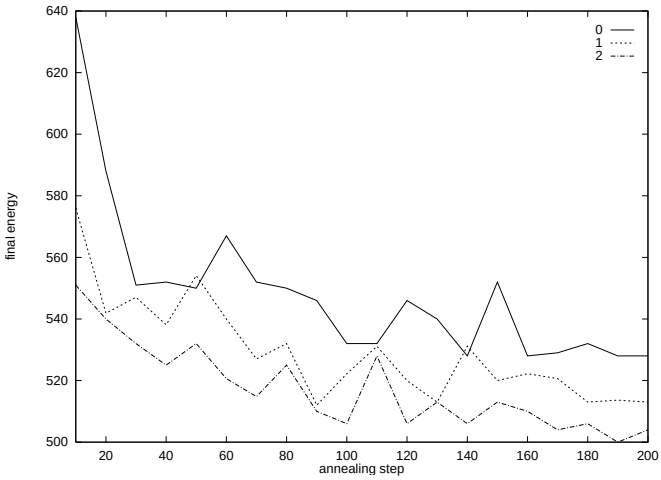


図 4.4: 最悪値の推移 (ランダム)

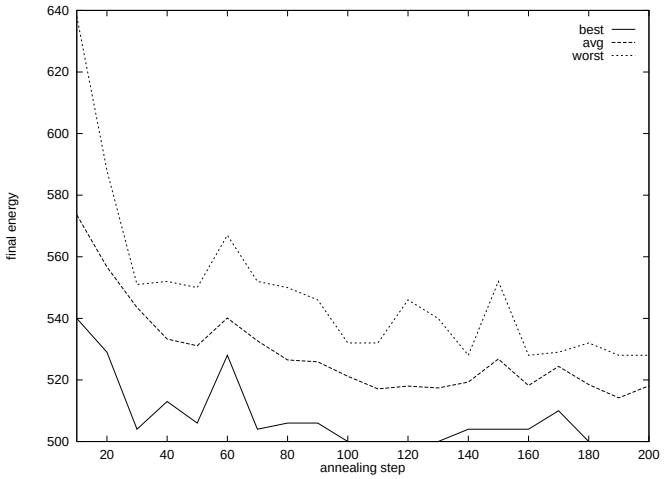


図 4.5: 固定モジュールのみの最良, 平均, 最悪値の推移 (ランダム)

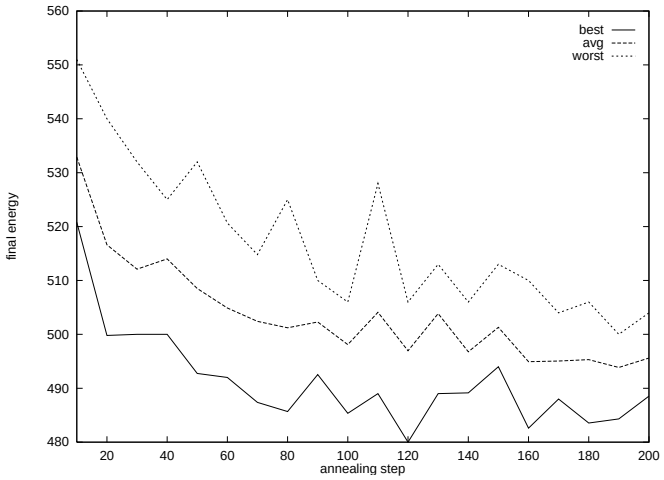


図 4.6: 合計面積一定モジュール 2 個の最良, 平均, 最悪値の推移 (ランダム)

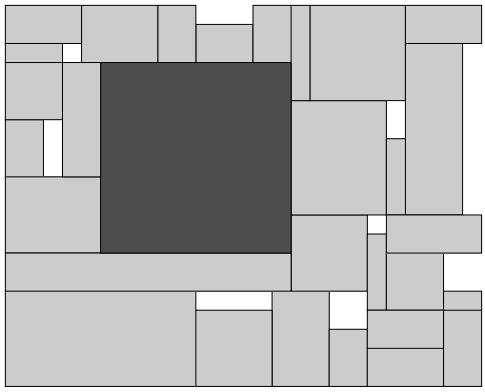


図 4.7: ソフトモジュールを含まない (ランダム)

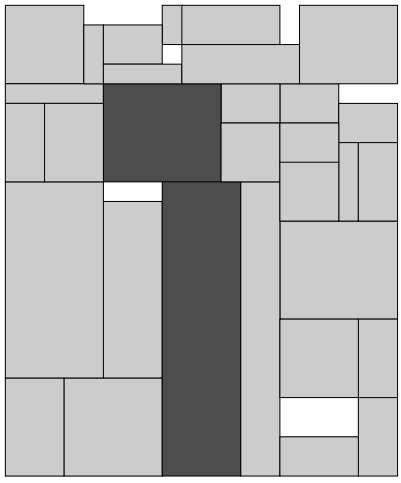


図 4.8: ソフトモジュールを 2 個含む (ランダム)

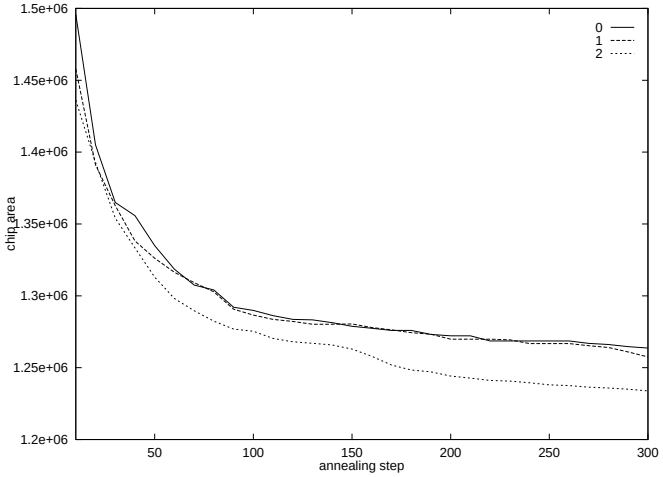


図 4.9: 平均値の推移 (ami33)

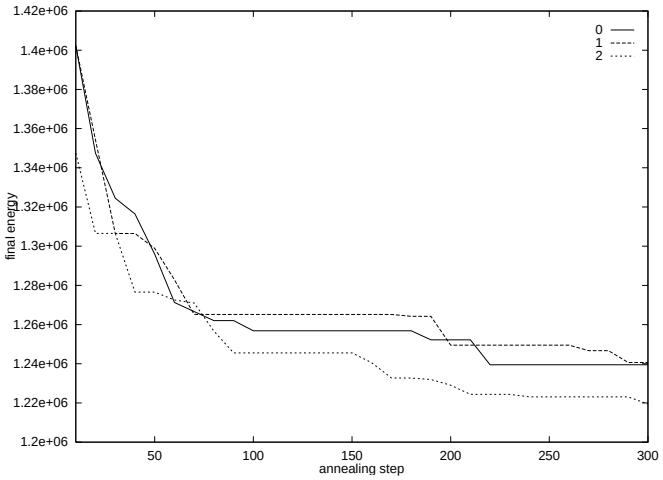


図 4.10: 最良値の推移 (ami33)

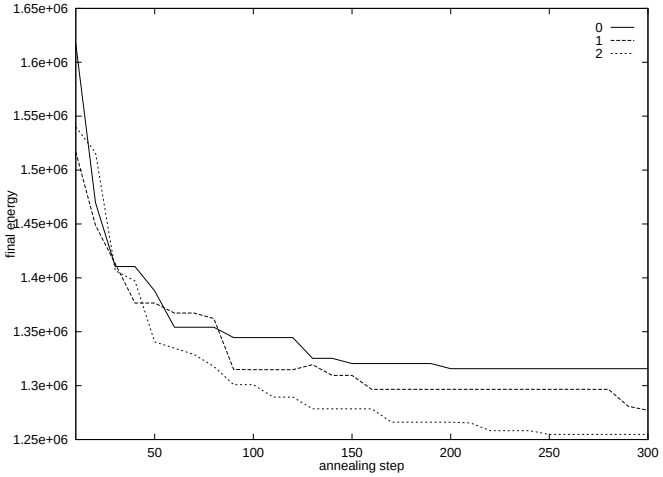


図 4.11: 最悪値の推移 (ami33)

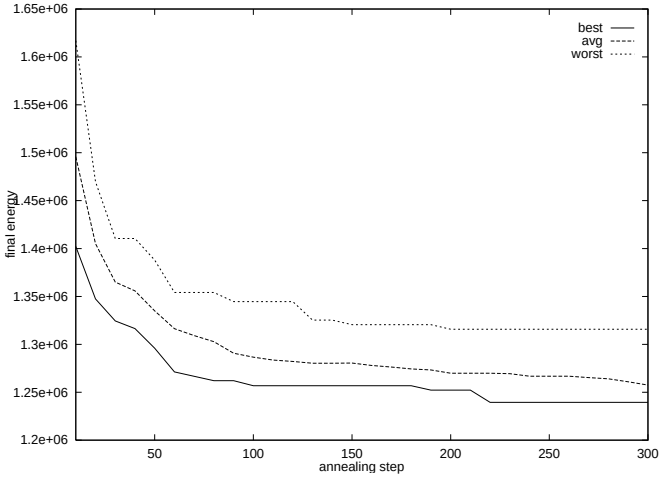


図 4.12: 固定モジュールのみの最良, 平均, 最悪値の推移 (ami33)

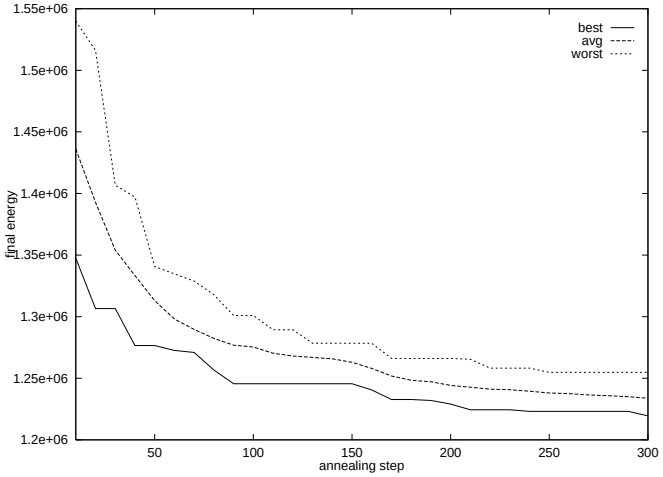


図 4.13: 合計面積一定モジュール 2 個の最良, 平均, 最悪値の推移 (ami33)

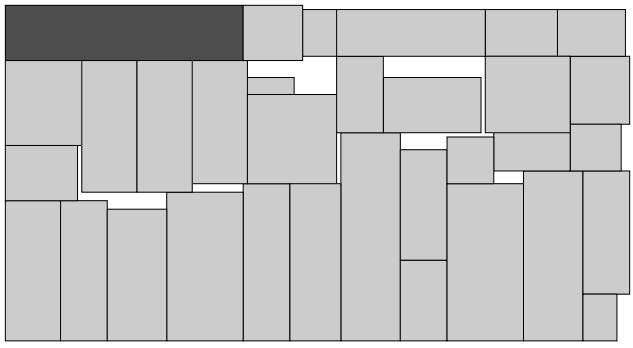


図 4.14: ソフトモジュールを含まない (ami33)

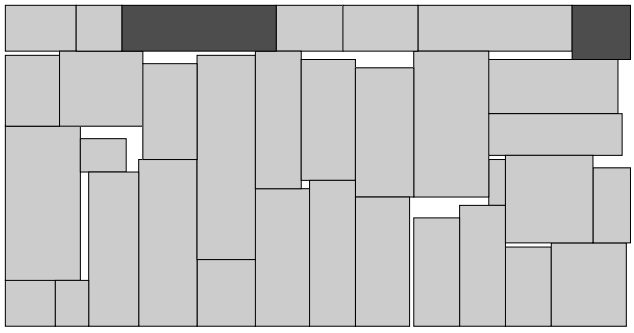


図 4.15: ソフトモジュールを 2 個含む (ami33)

第 5 章

結論

VLSI チップの集積度はますます増大しており，設計支援・自動化はその進歩に対応することが今後必要とされている．レイアウト設計の工程の一つである配置工程は，配置すべき矩形が色々な大きさである為に困難な問題であり，ソフトモジュールを取り扱う事はより困難な事である．チップ面積を最小にする書くソフトモジュールの形状を決定するフロアプラン領域最小化問題は様々な研究がされている．

従来研究では，指定された各ソフトモジュールの面積を保って，ソフトモジュールの幅と高さをチップ領域が最小になるように決定していた．しかしながら各ソフトモジュールの面積は，常に保存される必要はなく，ソフトモジュール面積の合計を維持すれば良いと場合がある．例えば，ランダムロジックでは通常モジュールを分割する事が認められ，その分割によりチップ面積をより小さくする事ができる可能性がある．本論文では，そのようなソフトモジュールを含む新しいフロアプラン最小化問題を定式化しそして解析的に解く方法を考えた．

それは，最大パス長の候補の性質から得た定理により，次の様に得られる．ソフトモジュール数分の最大パス長の候補を各制約グラフから選択すると，各ソフトモジュールの幅 (高さ) をチップの幅 (高さ) の線形関数で解く事ができ，その結果として最小化する為の一変数に関する 2 次有理関数とその変数の定義域を得る事ができる．

そして，提案手法を実装する事でその効果を確かめた所，実際にチップ面積は小さくなり，さらに，それは常に良い評価であるものが多く得る事ができた．一方で計算時間は，約 3.3 倍になる事もわかった．

今後の課題として，合計面積一定モジュール数増加に伴う計算時間の削減である．

謝辞

本研究の機会を賜わり，御助言，御指導して下さいました東京工業大学・電気電子工学科・梶谷洋司 教授に心から感謝の意を表します.

適切な御助言，御指導して下さいました北陸先端科学技術大学院大学情報科学研究科・金子峰雄 助教授，平石邦彦 助教授に感謝いたします.

多くの御助言を頂き，熱心に御指導して下さいました東京農工大学・電子情報工学科・藤吉邦洋 講師に深く感謝いたします.

本研究に関して，有益な御助言を頂きました東京工業大学・電気電子工学科・中武繁寿 助手，博士後期課程 3 年 泉知論 氏に感謝いたします.

また，常日頃からお世話になりました北陸先端科学技術大学院大学情報科学研究科博士後期課程 3 年 村田洋 氏をはじめとする北陸先端科学技術大学院大学・金子研究室，平石研究室の皆様に感謝いたします.

参考文献

[1] S. Wimmer and I. Koren and I. Cederbaum, "Floorplans, Planar Graphs, and Layouts", "IEEE Trans. on Circuit and Systems", vol. 35, Mar, 1988,pp. 267–278.

[2] K. Wang and W.-K. Chen, "Floorplan Area Optimization using Network Analogous Approach", Proc. IEEE International Symposium on Circuits and Systems, 1995, pp. 167–170.

[3] T.-S. Moh and T.-S. Chang and S. L. Hakimi, "Globally Optimal Floorplanning for a Layout Problem", journal = "IEEE Trans. on Circuit and Systems – I: Fundamental Theory and Applications", vol. 43, Sep, 1996,pp.713–720.

[4] R. H. J. M. Otten, "Automatic Floorplan Design", Proc. 19th ACM/IEEE Design Automation Conf., 1982,pp. 261–267.

[5] R. H. J. M. Otten, "Efficient Floorplan Optimization", Proc. ICCD, 1983,pp.499–503.

[6] L. Stockmeyer, "Optimal Orientations of Cells in Slicing Floorplan Designs", Information and Control, vol. 57,1983,pp.91–101.

[7] K.-H. Yeap and M. Sarrafzadeh, "An Integrated Algorithm for Optimal Floorplan Sizing and Enumeration", European Design Automation Conf.,1993,pp. 29–33.

[8] S. Wimer and I. Koren and I. Cederbaum, "Optimal Aspect Ratios of Building Blocks in VLSI", IEEE Trans. on Computer-Aided Design, vol. 8,Feb,1989,pp.139–145.

[9] P. Pan and C. L. Liu, "Area Minimization for Floorplans", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol.14,Jan, 1995,pp.123-132.

[10] D. F. Wong and P. S. Sakhamuri, "Efficient Floorplan Area Optimization", Proc. 26th ACM/IEEE Design Automation Conf., 1989,pp.586–589.

[11] T.-C. Wang and D. F. Wong, "Optimal Floorplan Area Optimization", IEEE Trans. on Computer-Aided Design, vol. 11,Aug,1992,pp.992–1002.

[12] K. Chong and S. Sahni, "Optimal Realizations of Floorplans", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 12,Jun,1993,pp.793–801.

[13] T.-C. Wang and D. F. Wong, "A Note on the Complexity of Stockmeyer's Floorplan Optimization Technique", Algorithmic Aspects of VLSI Layout, editor M. Sarrafzadeh and D. T. Lee, World Scientific, Singapore, 1992,pp.309–320.

[14] C.-H. Chen and I. G. Tollis, "Area Optimization of Spiral Floorplans", Journal of Circuits, Systems, and Computers, vol.3, 4,1993,pp.833–857.

[15] P. Pan and W. Shi and C. L. Liu, "Area Minimization for Hierarchical Floorplans", "Algorithmica",vol.15,1996,pp. 550–571.

[16] H.Murata, K.Fujiyoshi, S.Nakatake and Y.Kajitani, "Rectangle-Packing-Based Module Placement", Proc. International Conf. on CAD,1995,pp.472-479.

[17] S.Nakatake, K.Fujiyoshi, H.Murata, and Y.Kajitani, "Module Placement on BSG-Structure and IC Layout Applications," Proc. International Conf. on CAD,1996,pp.484-491.

付録

[定理.1] チップ全体の横方向における長さ X が $\mathcal{W}^c_{\{\}}$ よりも大きい最適解の中には，n 以上の本数の水平方向の候補パスの長さが X と等しいものが必ず存在する．

[証明.1] 最適解（チップ面積を最小とする解）の中で，次に述べる評価がもっとも良い最適解を考える

最適解の中で特に w_1 を最大とする解を考え，そのときの w_1 の値を w_1^{opt} とする．更に，上記の条件の下で w_2 を最大とする解を考え，そのときの w_2 の値を w_2^{opt} とする．更に，上記の条件の下で w_3 を最大とする解を考え，そのときの w_3 の値を w_3^{opt} とする．$\ldots$，更に，上記の条件の下で w_n を最大とする解を考え，そのときの w_n の値を w_n^{opt} とする．この最適解 $\left(w_1^{opt}, w_2^{opt}, \cdots, w_n^{opt}\right)$ は n 以上の本数の水平方向の候補パスの長さが X と等しいものが必ず存在する事を，以下では背理法で証明する為に，$k < n$ である k 本の独立な候補パスしかチップ全体の水平方向の長さ X に等しくないと仮定する．k 本の候補パスについては，式 (5.1) にて表すことができる．

$$\begin{pmatrix} X \\ X \\ \vdots \\ X \end{pmatrix} = \begin{pmatrix} b_1^1 & b_2^1 & \ldots & b_n^1 \\ b_1^2 & b_2^2 & \ldots & b_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ b_1^k & b_2^k & \ldots & b_n^k \end{pmatrix} \begin{pmatrix} w_1^{opt} \\ w_2^{opt} \\ \vdots \\ w_n^{opt} \end{pmatrix} + \begin{pmatrix} \mathcal{W}^c_{S^1} \\ \mathcal{W}^c_{S^2} \\ \vdots \\ \mathcal{W}^c_{S^k} \end{pmatrix} \tag{5.1}$$

$$\text{但し}, b_i^j = \begin{cases} 0 & \text{if } s_i \notin S^j \\ 1 & \text{if } s_i \in S^j \end{cases}$$

ここで，式 (5.1) における w_i^{opt} の値を変数 w_i に置き換えた式 (5.2) について考える．

$$\begin{pmatrix} X \\ X \\ \vdots \\ X \end{pmatrix} = \begin{pmatrix} b_1^1 & b_2^1 & \ldots & b_n^1 \\ b_1^2 & b_2^2 & \ldots & b_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ b_1 & b_2 & \ldots & b_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} + \begin{pmatrix} \mathcal{W}^c_{S^1} \\ \mathcal{W}^c_{S^2} \\ \vdots \\ \mathcal{W}^c_{S^k} \end{pmatrix} \tag{5.2}$$

$$\text{但し}, b_i^j = \begin{cases} 0 & \text{if } s_i \notin S^j \\ 1 & \text{if } s_i \in S^j \end{cases}$$

すると，変数 w_i の内の幾つかは式 (5.2) により一意の値に決まる（明らかに，w_i^{opt} と等しい値となる）．そして，残りの変数の値は不定解になるが，b_i^j の行列の rank 値が k であ

ることから．少なくとも $n-k$ 以上の変数が不定解になる．

不定解である変数 w_i の中で最も添字の小さいものを $w_{i'}$ とする．

候補パスの式は線形である事から式 (5.1) の $w_{i'}^{opt}$ を微小量 δ 増加させても，他の不定解である変数だけを微小変化させて，式 (5.3) を成り立たせることができる．

ここで，補題 1 から，不定解である変数 w_i に対応する w_i^{opt} は全て正の値（>0）であることから，δ の値を十分に小さく選べば，式 (5.3) の $w_i^{opt}+\delta, \cdots, w_k^{opt}-delta_1, \cdots, w_l^{opt}-deltas$ の値は全て 0 以上にでき，また，「k 本の独立な候補パスしかチップ全体の水平方向の長さ X に等しくない」という条件を満たしたままにできる．つまり，式 (5.2) と同様に，式 (5.3) で表される k 本の独立な候補パスも，最適解の一つである．

$$\begin{pmatrix} X \\ X \\ \vdots \\ X \end{pmatrix} = \begin{pmatrix} b_1^1 & b_2^1 & \dots & b_n^1 \\ b_1^2 & b_2^2 & \dots & b_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ b_1^k & b_2^k & \dots & b_n^k \end{pmatrix} \begin{pmatrix} w_1^{opt}(\text{一意}) \\ w_2^{opt}(\text{一意}) \\ \vdots \\ w_{i'}^{opt}+\delta \\ \vdots \\ w_d^{opt}+\delta_1 \\ \vdots \\ w_n^{opt}-\delta_s \end{pmatrix} + \begin{pmatrix} \mathcal{W}_{S^1}^c \\ \mathcal{W}_{S^2}^c \\ \vdots \\ \mathcal{W}_{S^k}^c \end{pmatrix} \tag{5.3}$$

$$\text{但し, } b_i^j = \begin{cases} 0 & \text{if } s_i \notin S^j \\ 1 & \text{if } s_i \in S^j \end{cases}$$

ここで，w_i^{opt} から δ だけ増加させた時に，与えられた合計面積 $\mathcal{S}$ と等しいならば，最適解の仮定に矛盾する．

δ だけ増加させた時，与えられた合計面積 $\mathcal{S}$ より大きくなる時，行列 (5.1) の X と $w_{1_{opt}}, w_{2_{opt}}, \cdots, w_{n_{opt}}$ の以外の値を変えずに，w_1 から w_n までの不定解の全てを微小減少させる事で，X をより小さくする事ができる．つまり注目している最適解よりも良い解が存在することにより矛盾する．

δ だけ増加させた時，与えられた合計面積 $\mathcal{S}$ より小さくなる時，候補パスが線形である事から，逆に δ を減少さえれば，合計面積を大きくでき，前者の場合と同様に矛盾する．

垂直制約グラフについても同様の事が言える．■

[補題.1] 行列 (5.2) において不定解となる w_j は，行列 (5.1) では，正の値を持つ．

[証明] 不定解 w_j が 0 であると仮定する．w_j を含む候補パスが行列 (5.1) の中に存在しないならば，$w_j=0$ ではなく，値を増加させる事ができ，仮定に矛盾する．行列 (5.1) の中

に存在する時，w_jの候補パスを

$$w_{i_{opt}} + \cdots + w_{j_{opt}} + \cdots + w_{k_{opt}} + \mathcal{W}^c_{\{s_{i,\cdots,j,\cdots,k}\}} \tag{5.4}$$

とする．この時，m_jを除いた候補パスは

$$w_{i_{opt}} + \cdots + w_{k_{opt}} + \mathcal{W}^c_{\{s_{(i,\cdots,k)}\}} \tag{5.5}$$

である．$w_j = 0$ であるので，式 (5.4) は，

$$w_{i_{opt}} + \cdots + w_{j_{opt}}(= 0) + \cdots + w_{k_{opt}} + \mathcal{W}^c_{\{s_{i,\cdots,j,\cdots,k}\}} \tag{5.6}$$

となる．式(5.5) と式(5.6) は，定数部のみが，違い残りは全て同じである．また，$\mathcal{W}^c_{\{s_{i,\cdots,j,\cdots,k}\}} \leq \mathcal{W}^c_{s_{i,\cdots,k}\}}$の関係があるので，式 (5.5) もまた，行列 (5.1) に存在する．行列 (5.1) の存在すると言う事は行列 (5.2) にも存在すると言う事である．ここで，行列 (5.2) で，式 (5.4), (5.5) について考えると，それぞれ，次の式になる．

$$w_i + \cdots + w_j + \cdots + w_k + \mathcal{W}^c_{\{s_{i,\cdots,j,\cdots,k}\}} \tag{5.7}$$

$$w_i + \cdots + w_k + \mathcal{W}^c_{\{s_{i,\cdots,kc}\}} \tag{5.8}$$

この式 (5.7),(5.8) は X と等しいので，計算すると $w_j = \mathcal{W}^c_{\{s_{s,\cdots,t}\}} - \mathcal{W}^c_{\{s_{s,\cdots,j,\cdots,t}\}}$となる．定数部によって w_jを表す事ができるので，w_jは，不定解ではなく一意の解を持つ事を示している．つまり，不定解を選んだと言う事に矛盾する．不定解は常に正である事が言える．
■