

Title	座標固定モジュールを扱うBSG構造におけるモジュール配置手法の考案
Author(s)	古屋, 正浩
Citation	
Issue Date	1997-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1029
Rights	
Description	Supervisor:平石 邦彦, 情報科学研究科, 修士

修 士 論 文

座標固定モジュールを扱う BSG 構造における モジュール配置手法の考案

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

古屋 正浩

1997 年 2 月 14 日

目次

1	はじめに	1
2	2次元パッキング問題	3
2.1	産業界からの要求	3
2.2	諸定義と定式化	6
3	Bounded Sliceline Grid	7
4	配置座標指定を許す2次元矩形パッキング問題	12
4.1	p-モジュールの扱いにおける困難	12
4.2	f-モジュールの計算順序	14
4.3	p-モジュールを扱う写像アルゴリズム	16
5	レクトリニアモジュールの配置	19
5.1	lead-モジュールと準p-モジュール	19
5.2	レクトリニアモジュールを扱う写像アルゴリズム	21
6	実験	24
6.1	シミュレーティッドアニーリングへの適用	24
6.2	p-モジュールを含むパッキング	25
6.3	レクトリニアモジュールを含むパッキング	27
7	おわりに	28

第 1 章

はじめに

近年の集積回路 (IC) やプリント基板 (PCB) 設計における設計仕様の複雑化や回路規模の増大に対処するために、レイアウト設計自動化技術における飛躍が期待されている。

レイアウト設計は、主に配置と配線の工程からなり、特にモジュール配置問題は、簡単には、小矩形の集合を重ねたりなくできるだけ小さな矩形領域に配置するという、2 次元矩形パッキング問題として捉えることができる。

最近の研究成果により、モジュール配置問題における高度な設計要求を満たす 2 次元矩形パッキング技術として、メタグリッド方式が提案された。これらの論文において、メタグリッドと呼ばれる概念とそれを実現する 2 つのメタグリッド SEQ-PAIR, BSG が紹介されている。

このメタグリッドは、一切の物理的な座標情報を持たず、矩形の相対的な位置関係のみにより、2 次元矩形パッキングの解を表現することができる。村田ら [1] は、メタグリッド SEQ-PAIR により面積最小解 (最適解) を含みながらも、解の数が有限であるような解空間の構成について述べている。このメタグリッドにより構築される解空間に対し、シミュレーティッドアニーリングのような確率的探索手法を適用することにより、高品質な解が得られることが実験的に示されている。

また、中武ら [2] は、BSG と呼ばれる新しいメタグリッドを提案し、PCB やアナログ IC を対象としたいいくつかの応用技術を紹介した。しかしながら、このメタグリッドの歴史は浅く、実際回路に適用するためには、さらに多くの応用技術の開発が必要である。

本研究においては、メタグリッド BSG による 2 次元矩形パッキング技術を基本技術とする PCB 設計の部品配置システムの開発を行う。

PCB 設計自動化においては、設計者により配置座標の指定される座標固定部品の取り扱いが要求される [3]。しかしながら、従来のメタグリッドから 2 次元矩形パッキング解への写像アルゴリズムは、各部品の垂直水平座標を独立に計算するため、このような座標固定部品の取り扱いが困難であった。

本論文では、メタグリッド BSG において、BSG により表現される解空間の性質を変え、各部品の座標を垂直水平同時に決定する写像アルゴリズムを示す。この写像アルゴリズムを用いて、座標固定部品を扱う手法を提案する。

さらに、PCB 設計では、非矩形であるレクトリニア部品の取り扱いも重要である [4, 5, 6]。中武ら [2] も BSG を用いた L 型部品の取り扱いを提案しているが、本論文では、座標固定部品の取り扱い手法を拡張させることにより、一般レクトリニア部品を 2 次元パッキングするヒューリスティックの提案を行う。

実験においては、座標固定部品の数によらず、高品質な結果を得ることに成功した。また、実際の PCB データへの適用を行った結果、座標固定を行わない場合と同等の解を得ることができた。さらに、複雑な構造のレクトリニア部品を含んだ入力に対しても高品質な解を得ることができた。

第 2 章

2 次元パッキング問題

本章では, 本研究における問題の定式化と諸定義を行なう.

2.1 産業界からの要求

一般に, IC や PCB の配置問題は, 簡単には, 様々な大きさの矩形モジュールを重なりなくできるだけ小さな矩形領域に配置する, という 2 次元矩形パッキング問題として捉えることができる (図 2.1 参照).

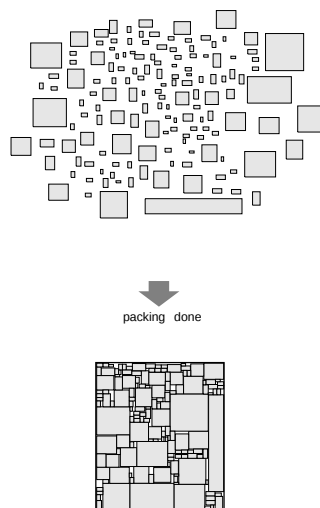


図 2.1: 2 次元矩形パッキング問題

しかし、実際の IC や PCB 設計では、VLSI 設計におけるマクロセルや PCB におけるコネクタのように、設計者により配置座標の指定されているモジュールがある。(図 2.2 参照).

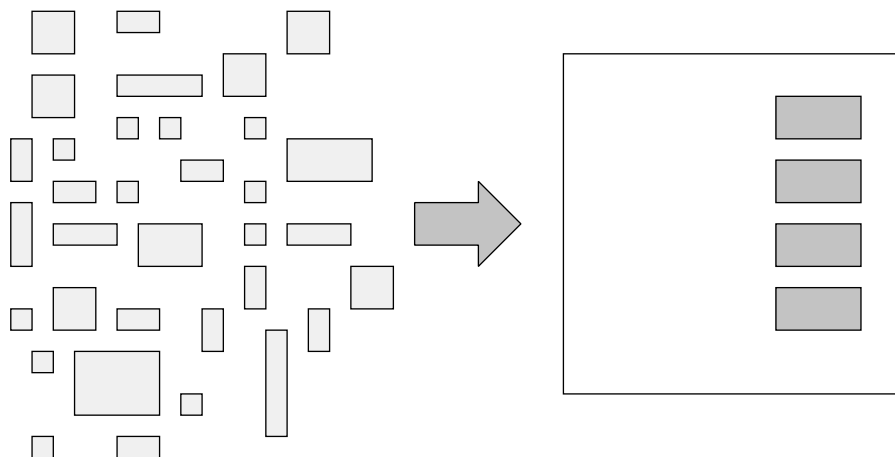


図 2.2: 既配置部品

さらに、複数の素子や部品をモジュール化して扱うことがあるため、モジュールの形状は矩形に限られない。このようなモジュールは、垂直水平線分により囲まれる閉領域、すなわちレクトリニア領域により表現される(図 2.3 参照)。矩形領域はレクトリニア領域の特殊形である。

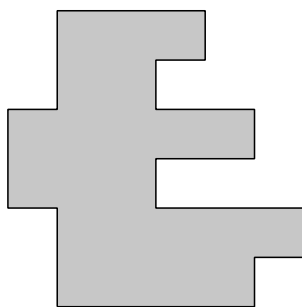


図 2.3: レクトリニアモジュール

本研究においては, 2次元矩形パッキング問題に制限を加えた, 以下のような問題を考える.

2次元矩形パッキング問題

入力: 矩形モジュール

出力: モジュールの重なりのない配置

最適化目標: 全てのモジュールを囲む領域の面積最小化



座標指定部品を許す 2次元矩形パッキング問題

入力: 配置座標を指定された矩形モジュールを含む矩形モジュール集合

出力: モジュールの重なりのない配置

最適化目標: 全てのモジュールを囲む矩形領域の面積最小化



座標固定部品を許す 2次元レクトリニアパッキング問題

入力: 配置座標の指定されたレクトリニア形状のモジュールを含むレクトリニア形状のモジュール集合

出力: モジュールの重なりのない配置

最適化目標: 全てのモジュールを囲む矩形領域の面積最小化

2次元矩形パッキング問題は, 座標指定部品を許す 2次元矩形パッキング問題の部分問題である. また, 座標指定部品を許す 2次元矩形パッキング問題は, 座標指定部品を許す 2次元レクトリニアパッキング問題の部分問題である. そこで今後は, 座標指定部品を許す 2次元レクトリニアパッキング問題を単に 2次元パッキング問題と呼ぶものとする.

まず, 用語の定義と 2次元パッキング問題の定式化を行なう.

2.2 諸定義と定式化

- 形状が指定されたレクトリニア領域をモジュールと呼ぶ.
- モジュールの座標は, モジュールを囲む最小矩形の左下角の座標で表す.
- 座標が指定されているモジュールを, p-モジュール (pre-placed-module) と呼ぶ.
- 座標が指定されていないモジュールを, f-モジュール (floating-module) と呼ぶ.
- 形状が矩形であるモジュールを矩形モジュール, 形状が矩形以外のモジュールをレクトリニアモジュールと呼ぶ.
- (モジュールの) パッキングとは, モジュールの重ならない配置とする.
- 2次元パッキング問題

入力: 座標が指定されているモジュールを含むモジュール集合

出力: モジュール集合のパッキング

最適化目標: モジュール集合を囲む領域の面積最小化

第 3 章

Bounded Sliceline Grid

2 次元矩形パッキング問題では, 既公表の Bounded Sliceline Grid(BSG) に基づく配置方法で高品質な解を得ることが可能である. 本研究においては, その応用技術の幅広さと将来性を考え, 2 次元パッキング問題を解く基本技術としてこの BSG を採用した.

BSG 構造について文献 [2] の必要な部分を再述する. 尚, 本章でのモジュールは矩形モジュールとする.

BSG は, モジュールの一切の物理座標を持たない, 位相的な位置関係のみで配置を表現するデータ構造であるが, 便宜上, BSG の構造を xy 座標系を用いて次のように定義する.

BSG は BSG-unit(又は単に unit) と呼ばれる線分で構成する. unit 集合 U_{BSG} は, 以下に定義する線分 $H_{i,j}, V_{i,j}$ (i, j は整数) により構成する (図 3.1(A) 参照).

$$U_{BSG} = \{V_{i,j} | i + j : \text{even}\} \cup \{H_{i,j} | i + j : \text{odd}\}$$

$$H_{i,j} = \{(x, y) | i - 1 < x < i + 1, y = j\}$$

$$V_{i,j} = \{(x, y) | x = i, j - 1 < y < j + 1\}$$

i, j は線分 $H_{i,j}, V_{i,j}$ 上の中心座標を表している. $H_{i,j}, V_{i,j}$ はそれぞれ水平 unit, 垂直 unit と呼ぶ.

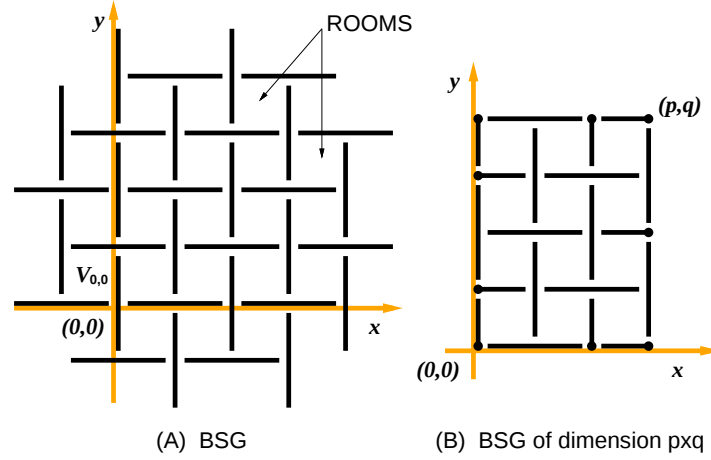


図 3.1: Bounded Sliceline Grid

2つの垂直 unit $V_{i,j}, V_{i',j'}$ において, $|i' - i| = 1$ かつ $|j' - j| = 1$ のとき, $V_{i,j}$ と $V_{i',j'}$ は隣接関係にある (又は, 隣接している) といい, $i' - i = 1$ かつ $|j' - j| = 1$ のとき $V_{i',j'}$ は $V_{i,j}$ の右側関係にある (又は, 右にある) という. 垂直 unit V_a の右に別の垂直 unit V_b があり, V_b の右に V_c がある場合, V_c も V_a の右にあるという. 水平 unit に関しても, 隣接関係と上側関係を同様に定義する.

また, 隣接する垂直, 水平 unit の組によって囲まれる矩形領域をルームと呼ぶ.

定義より BSG は無限グリッド空間であるが, p 行 q 列のルームを持つ有限グリッド空間 $BSG_{p \times q}$ を作業平面とする. このとき, $BSG_{p \times q}$ の左下角が原点, 右上角が (p, q) となる. $BSG_{p \times q}$ の周囲の unit で外周から突き出す部分は切り取られる. 例えば $V_{0,0}$ は $\{(x, y) | x = 0, 0 \leq y < 1\}$ となる (図 3.1(B) 参照).

i 行目 j 列目のルームを $r_{i,j}$ と記す.

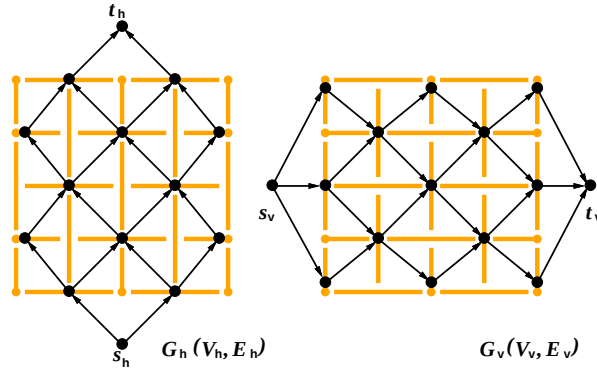


図 3.2: unit 隣接グラフ

さらに, unit の上下関係を表現する水平 unit 隣接グラフ $G_h(V_h, E_h)$ を以下のように定義する.

領域 $BSG_{p \times q}$ において, $V_h = \{s_h, t_h\} \cup \{u_{i,j}\}$ とする. $u_{i,j}$ は水平 unit $H_{i,j}$ に対応する点である. ソース s_h から $u_{i,0}$, また, $u_{i,q}$ から シンク t_h に枝をつくる. さらに, $H_{i',j'}$ が $H_{i,j}$ に隣接し, かつ $H_{i,j}$ の上にあるとき, $u_{i,j}$ から $u_{i',j'}$ に枝をつくる. unit の左右関係を表現する垂直 unit 隣接グラフ $G_v(V_v, E_v)$ を同様に定義する. (図 3.2参照)

次に, BSG からパッキングへの写像アルゴリズムについて説明する. $BSG_{p \times q}$ の各ルームに対し高々1 モジュールを割り当てる. モジュールの割り当てられなかったルームは空ルームとする. G_h において, ソース, シンクに接続する枝集合には重み 0 が与えられる. それ以外の枝集合は, ルーム集合と 1 対 1 対応し, 対応するルームに割り当てられたモジュールの高さが重みとして与えられる. 空ルームの場合は重み 0 である. 同様に, G_v の枝にはモジュールの幅が与えられる.

ここで, 任意のモジュール m の割り当てられているルームを r とする. m の左下角の x 座標は, G_v におけるソースから r の左にある unit までの最大パス長である. 同様に m の y 座標は, G_h における ソースから r の下にある unit までの最大パス長である. BSG 上の割り当てからパッキングへの写像の計算例を図 3.3 に示す.

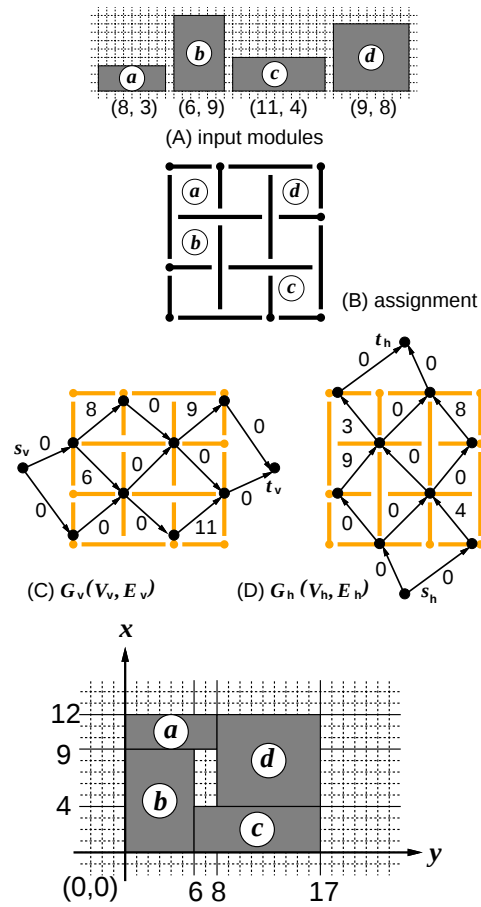


図 3.3: BSG からパッキングへの写像例

まとめれば：

モジュールを $BSG_{p \times q}$ に「各ルームに高々 1 個」という規則で割り当てれば一意に一つのパッキングが得られる．以上のことから BSG については以下のことが言える．

性質 1(計算複雑度) 各ルームまでの最大パス長は, unit 隣接グラフのソースからシンクまでの最大パスを求める過程において計算されるので, すべてのモジュールの配置を求める計算時間は, $O(pq)$ である.

性質 2(位相関係) BSG 上の任意の 2 つのルームには, 必ず上下, あるいは左右のどちらか一方の関係だけが存在する. 従って, BSG 上に割り当てられた任意の 2 つのモジュールには, 必ず上下, あるいは左右のどちらか一方の関係だけが存在する.

性質 3(極小性) unit 隣接グラフにより計算されるパッキングは, モジュール間の左右, 上下関係が同じパッキング集合の中で面積最小である.

性質 4(最適性) モジュール数を n とするとき, $BSG_{n \times n}$ への割当てにより構成される 2 次元矩形パッキングの解空間は, 面積最小解を含むことが証明されている.

さらに, 論文 [2] においては, 2 次元矩形パッキング問題を解くために, シミュレーティッドアニーリングなどの確率的探索手法を用いて, BSG 上でのモジュールの様々な割り当てに対するパッキングを評価し, 最良の解を選択している.

図 3.4 に BSG を用いたパッキングの例を示す.

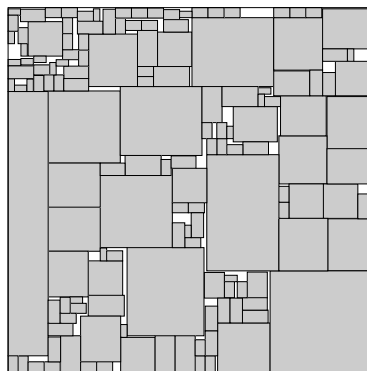


図 3.4: BSG による 146 モジュールのパッキング

第 4 章

配置座標指定を許す 2 次元矩形パッキング問題

本章では, BSG による 2 次元矩形パッキング手法を基本技術として配置座標指定を許す 2 次元矩形パッキング問題を解くアルゴリズムの提案を行う.

4.1 p-モジュールの扱いにおける困難

前章で説明したように, BSG を用いた 2 次元矩形パッキング手法は, すべてのモジュールを BSG 上に割り当て, 垂直, 水平 unit 隣接グラフを構成し, それぞれの最大パス長を計算することにより, パッキングを得るものであった.

しかしながら, 2 個以上の p-モジュールの BSG への割り当てを考えた場合, 座標指定の制約を満たすことができない割り当てが存在する. 言い替えれば, BSG により表現されるモジュール間の相対位置関係に矛盾する座標指定の制約は満たすことはできない. 例えば, 図 4.1 のように, b の指定座標が a の右にある p-モジュール a, b があったとする. このとき, a を b の右関係にあるルームに割り当てると指定座標の制約を同時に満たすことはできない.

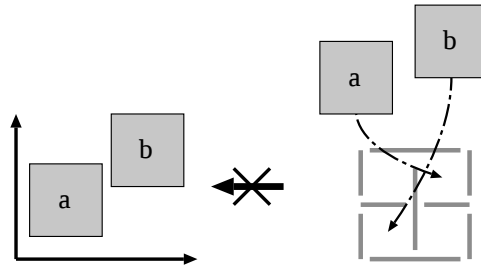


図 4.1: p-モジュールの位置関係による割り当ての困難

そこで，本提案手法においては，以下の方針により BSG からパッキングへの写像を行う．

1. まず，p-モジュールはあらかじめ指定座標に配置しておき，f-モジュールのみを BSG 上に割り当てる．
2. 次に，BSG 上の割り当てからパッキングへ写像するとき，f-モジュールと p-モジュールが重なりを生じないように f-モジュールの座標を決定する．

しかし，従来の写像アルゴリズムのように，モジュールの x, y 座標を垂直，水平 unit 隣接グラフにより独立に計算している限り，f-モジュールが p-モジュールに重なることを判定できない．

そこで，まず，写像アルゴリズムの途中で，割り当てられた f-モジュールの x, y 座標を同時に計算する手法を提案する．

4.2 f-モジュールの計算順序

BSG を用いて高品質なパッキングを探索するためには，前章における性質 1(計算複雑度) と性質 3(極小性) が重要である．そこで，本研究では，これらの性質を失うことなく，各 f-モジュールの x, y 座標を同時に決定する写像アルゴリズム Twist Mapping の提案を行う．

Twist Mapping

入力 : BSG 上での f-モジュールの割り当て

出力 : パッキング

1. すべての垂直, 水平 unit の x, y 座標を 0 にする．
2. $BSG_{p \times q}$ のルーム集合に対し, $r_{i,j}$ の系列 Π を次のように決定する．

$$\Pi = \Pi_0, \Pi_1, \dots, \Pi_{p+q-2}$$

$$\Pi_k = \begin{cases} r_{k,0}, r_{k-1,1}, \dots, r_{0,k} & (k : \text{even}) \\ r_{0,k}, r_{1,k-1}, \dots, r_{k,0} & (k : \text{odd}) \end{cases}$$

3. 系列 Π の示すルーム順にそのルームに割り当てられた f-モジュールの x, y 座標とルームの大きさを次のように計算する．

ルーム a において, a の左, 右の unit の x 座標と, 下, 上にある unit の y 座標をそれぞれ x_r, x_l と y_b, y_t とする．

4. (a) a が空ルームである場合:

$$x_r \leftarrow \max(x_r, x_l)$$

$$y_t \leftarrow \max(y_t, y_b)$$

に更新する．

- (b) a に f-モジュール m が割り当てられている場合:

m の左下角の x, y 座標を x_r, y_b に一致させる．

m の右上角の x, y 座標を x_m, y_m とすると

$$x_r \leftarrow \max(x_r, x_m)$$

$$y_t \leftarrow \max(y_t, y_m)$$

に更新する (図 4.2 参照) ．

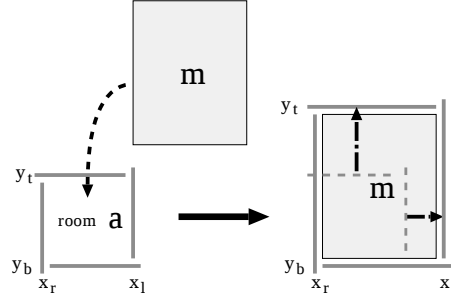


図 4.2: Twist Mapping の 4.(b) の計算

$BSG_{5 \times 5}$ のときの系列 Π を 図 4.3 に示す .

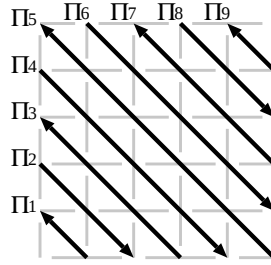


図 4.3: 系列 Π

定理 1 Twist Mapping は , 性質 1, 性質 3 を失うことなく , パッキングを生成できる .

証明 水平 unit 隣接グラフ G_h は , 非循環有向グラフであるため , 動的計画法により , ソースからシンクまでの最大パスを $O(pq)$ で計算できる . この計算過程において , 枝 $(u_{i,j}, u_{i',j'})$ を計算する前に , 点 $u_{i,j}$ に入るすべての枝について計算が完了していなければならない (条件 A) . G_v についても同様である .

また , G_h, G_v の定義より , ソース, シンクに接続しない枝集合はそれぞれルームと 1 対 1 対応するので , ルームの大きさの計算は , 枝の計算と等価である .

ここで , 任意の 2 つのルームを a, b とする . 系列 Π において , a が b よりも前にあるなら , a は b の左あるいは下にあるので , 系列 Π の示すルーム順に , ルームの大きさの計算を行うことにより , 条件 A が満たされる .

従って , G_v, G_h の最大パスが $O(pq)$ で計算できるので , 性質 1 , 性質 3 は満たされる .

□

4.3 p-モジュールを扱う写像アルゴリズム

系列 II の示すルーム順に , ルームに割り当てられているモジュールの座標を計算する際に , p-モジュールと f-モジュールの重なりが生じる .

そこで , f-モジュール m_f と p-モジュール m_p の重なりがある場合 , 重なりがなくなるまで , m_f を垂直あるいは水平に平衡移動させるアルゴリズム Shift F-Module を考える .

Shift F-Module

入力 : 重なりのある m_f, m_p の配置

出力 : 重なりのない m_f, m_p の配置

乱数により , m_f を垂直と水平方向のどちらに平行移動するか決定する .

1. 水平方向に移動する場合 (h-shift):

m_f の左下角の x 座標を m_p の右上角の x 座標に一致させる .

2. 垂直方向に移動する場合 (v-shift):

m_f の左下角の y 座標を m_p の右上角の y 座標に一致させる .

さらに , この Shift F-Module を用いて , すべての p-モジュールと f-モジュールを重なりなく配置するための写像アルゴリズム Twist Mapping with P-Module を提案する .

Twist Mapping with P-Module

入力 : BSG 上での f-モジュールの割り当てと

p-モジュールの配置

出力 : パッキング

1. すべての垂直, 水平 unit の x, y 座標を 0 にする .
2. Twist Mapping と同様に , $BSG_{p \times q}$ に対しルームの系列 Π を決定する .
3. 系列 Π の示すルーム順にそのルームに割り当てられた f-モジュールの x, y 座標とルームの大きさを次のように計算する .
ルーム a において , a の左, 右の unit の x 座標と下, 上の unit の y 座標をそれぞれ x_l, x_r と y_b, y_t とする .
4. (a) a が空ルームである場合:
$$x_r \leftarrow \max(x_r, y_l)$$
$$y_t \leftarrow \max(y_t, y_b)$$
 に更新する .
(b) a に f-モジュール m が割り当てられている場合:
 m の左下角の x, y 座標を x_l, y_b に一致させる .
このとき , m と 重なる p-モジュールが存在する場合には ,
Shift F-Module を繰り返し重なりを除去する .
 m の右上角の x, y 座標を x_m, y_m とすると
$$x_r \leftarrow \max(x_r, x_m)$$
$$y_t \leftarrow \max(y_t, y_m)$$
 に更新する .

図??は Twist Mapping with P-Module における f-モジュールと p-モジュールに重なりが生じたときの f-モジュールの配置の決め方の一例を示す .

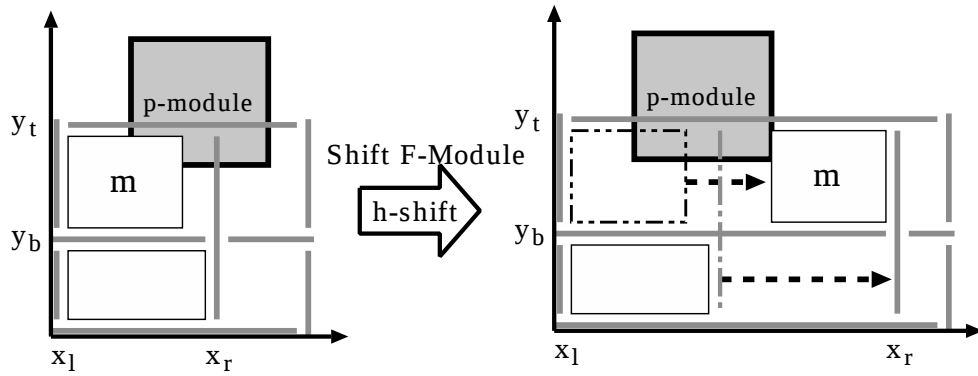


図 4.4: Twist Mapping with P-Module のルームの大きさの計算

f-モジュールと p-モジュールは，アルゴリズムの定義より，明らかに重ならない．p-モジュール同士は，入りに重なりがない場合には，出力においても重なりはない．さらに，各 f-モジュールは別々のルーム内に配置される．このため任意の 2 つの f-モジュール間に，左右あるいは上下関係のどちらかが存在する．従って，f-モジュール同士の重なりは生じない．

また，Shift F-Module の計算複雑度が $O(M^2)$ であるため，モジュール数を M としたとき，Twist Mapping with P-Module の計算複雑度は， $O(pqM^2)$ である．

第 5 章

レクトリニアモジュールの配置

実際の IC , PCB 設計においては , 複数のモジュールの相対位置を維持したまま配置したいという要求がある . このためレクトリニアモジュールの取り扱いが重要となる .

本章においては , 配置座標指定のあるレクトリニアモジュールを単に p-モジュールと呼び , 指定のないレクトリニアモジュールを f-モジュールと呼ぶことにする .

また , レクトリニアモジュールは , 複数の矩形集合としてみなすことができる . レクトリニアモジュール l に対応する矩形集合を l の矩形集合 と呼ぶ .

5.1 lead-モジュールと準 p-モジュール

レクトリニアモジュール l の矩形集合において , 矩形を 1 つ選択し , l の lead-モジュール とする . また , l の矩形集合に含まれ , lead-モジュール以外の矩形を l の準 p-モジュール とする (図 5.1 参照) . l の準 p-モジュールの配置は , l の lead-モジュールの配置座標に従い決定される .

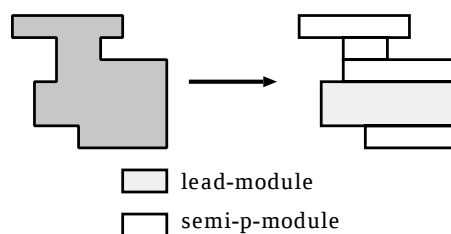


図 5.1: lead-モジュールと準 p-モジュール

本研究において提案するレクトリニアモジュールを扱う写像アルゴリズムは，次に示す方針に従う(図 5.2 参照)．

1. まず，レクトリニアモジュールを矩形分解し，lead-モジュールを選択する．
2. p-モジュールはあらかじめ指定座標に配置しておき，f-モジュールの lead-モジュールのみを BSG 上に割り当てる．
3. BSG 上の lead-モジュールの割り当てから，配置するレクトリニアモジュールの順序を決定し，すでに配置されているモジュールと重なりを生じないように，各レクトリニアモジュールの座標を決定する．

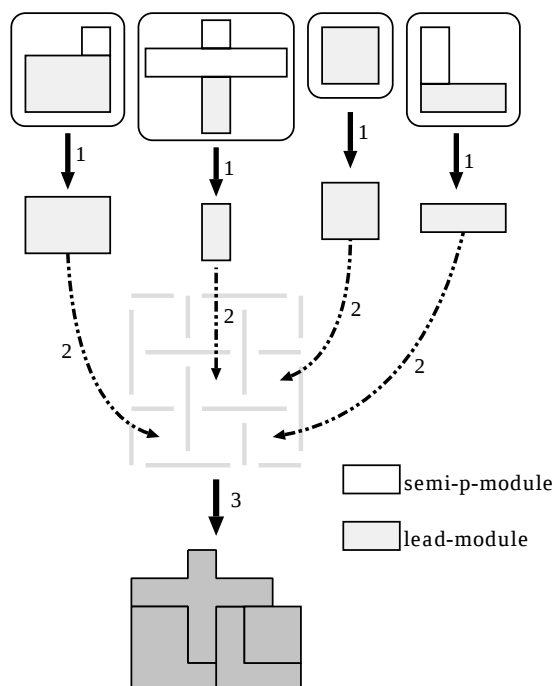


図 5.2: レクトリニアモジュールの写像アルゴリズム

5.2 レクトリニアモジュールを扱う写像アルゴリズム

BSG 上の lead-モジュールの割り当てからパッキングへの写像アルゴリズムを説明する前に、まず、p-モジュールである l_p と f-モジュールである l_f の配置に重なりを除去するアルゴリズム Shift Rectilinear を導入する。

Shift Rectilinear

入力 : 重なりのある l_p, l_f の配置

出力 : 重なりのない l_p, l_f の配置

l_f と l_p の各矩形集合においてすべての矩形間の重なりがなくなるまで、以下の処理を繰り返す。

1. l_f と l_p の各矩形集合において、重なりのある 2 つのモジュール をそれぞれ m_f, m_p とする。
2. m_f, m_p に Shift F-Module を適用し、重なりを除去する。
3. m_f に従い、 l_f の矩形集合の x 座標を更新する。

図 5.3 に、Shift Rectilinear の動作例を示す。

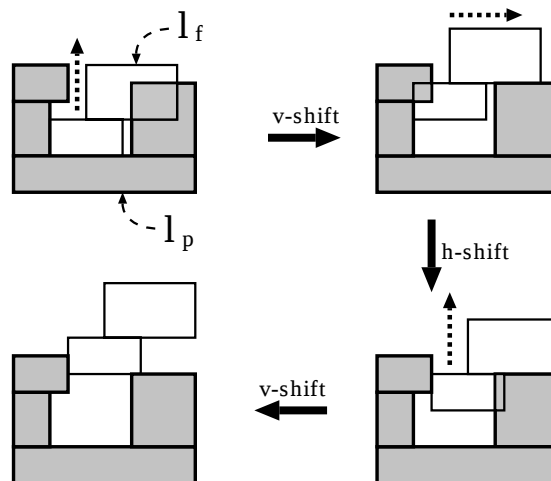


図 5.3: Shift Rectilinear の動作例

この Shift Rectilinear を用いて , BSG 上でレクトリニアモジュールを扱う写像アルゴリズム Twist Mapping with Rectilinear を提案する .

Twist Mapping with Rectilinear

入力 : BSG 上での f-モジュールの
lead-モジュールの割り当てと
p-モジュールの配置

出力 : パッキング解

1. すべての垂直, 水平 unit の x, y 座標を 0 にする .
2. Twist Mapping と同様に , $BSG_{p \times q}$ に対しルームの系列 Π を決定する .
3. 各レクトリニアモジュールを矩形集合に分解し , 左辺と下辺が他の矩形と隣接しない矩形を 1 つ選び , lead-モジュールとする .
4. 系列 Π の示すルーム順にそのルームに割り当てられた lead-モジュールの x, y 座標とルームの大きさを次のように計算する .
ルーム a において , a の左, 右の unit の x 座標と下, 上の unit の y 座標をそれぞれ x_l, x_r と y_b, y_t とする .
5. (a) a が空ルームである場合:
$$x_r \leftarrow \max(x_r, y_l)$$
$$y_t \leftarrow \max(y_t, y_b) \text{ に更新する .}$$

(b) a に lead-モジュール m が割り当てられている場合:
 m の属するレクトリニアモジュールを l_m とし , m の左下角の x, y 座標を x_l, y_b に一致させる .
このとき , l_m の矩形集合と p-モジュールが存在する場合は , Shift Rectilinear を繰り返し重なりを除去する .
 m の右上角の x, y 座標を x_m, y_m とすると
$$x_r \leftarrow \max(x_r, x_m)$$
$$y_t \leftarrow \max(y_t, y_m) \text{ に更新する .}$$

 m に従い l_m の矩形集合の座標を決定し , l_m を p-モジュールとする .

Twist Mapping with Rectilinear においては，既配置のモジュールを p-モジュールとして扱い，配置を行うモジュールとの重なりを逐次調べるため得られるパッキングにおいて重なりは生じない．

また，レクトリニアモジュールが最大 L 個の矩形により構成されるとき，レクトリニアモジュール数を M とすると、Shift Rectilinear の計算複雑度は $O(M^2L^3)$ となる．従って，Twist Mapping with Rectilinear の計算複雑度は， $O(pqM^2L^3)$ となる．

第 6 章

実験

前章までにおいて， p -モジュールやレクトリニアモジュールを含む場合の BSG からパッキングへの写像アルゴリズムを示した．

本章では，これらの写像アルゴリズムをシミュレーティッドアニーリングに適用し，配置座標の指定のある 2 次元レクトリニアパッキング問題を解くアルゴリズムの提案を行う．

入力となるデータは，ランダムに大きさを作成したデータと実際の PCB の製品から抽出したデータを用いた．

6.1 シミュレーティッドアニーリングへの適用

提案するパッキングアルゴリズムを，シミュレーティッドアニーリングに適用する写像アルゴリズムにより分類し，表 6.1 に示す．

各シミュレーティッドアニーリングにおいて，BSG のルームの中身の対交換とモジュールの 90° 回転， x 軸反転によりパッキングの隣接解が生成される．また，評価関数はすべてのモジュールを囲む最小矩形の面積（基板面積）とし，アニーリングスケジュールは標

表 6.1: 提案パッキングアルゴリズム

Packing Algorithm	Mapping Algorithm
$SA_{p-module}$	Twist Mapping with P-Module
$SA_{rectilinear}$	Twist Mapping with Rectilinear

準的なものを利用した。

尚、プログラムはいずれも Sun-OS 上に C 言語で実装した。

6.2 p-モジュールを含むパッキング

まず、 $SA_{p-module}$ における p-モジュールの数の変化に対する基板面積の影響を調べるために、200 個の矩形モジュールの内、p-モジュールの数が 0,2,5,10,20,40,60,80,100 の場合について実験した。用意した 200 個の矩形モジュールは、高さと幅を実問題に即した範囲内でランダムに作成した。

実問題では、面積の大きなモジュールの座標を指定する傾向があるという熟練した PCB 設計者の経験を参考にし、指定したモジュールを囲む最小矩形の面積が 200 モジュールの合計面積以下になるように、面積の大きな順に p-モジュールとして指定した。

また、Shift F-Module において平行移動の選択を乱数に頼らない方法として、それぞれの方向に移動した場合の距離を比較し、短い方を選択するという変更を施した Shift F-Module' についても実験を行った。

図 6.1 は、Shift F-Module と Shift F-module' を用いた $SA_{p-module}$ に対し、3 種類の入力データによる実験結果の平均をグラフを示す。(A)(B) とも横軸は p-モジュールの数である。(A) の縦軸は入力モジュールの合計面積を 1 としたときのパッキングを囲む最小矩形の面積の比率である。(B) の縦軸は実行時間である。実験の結果から次のような観察を得る。まず、 $SA_{p-module}$ を用いることにより、p-モジュールの数によらず高品質なパッキングが得られることが示される。計算時間に関しては、p-モジュール数が全体の半分以下では線形に増大する。これは、f-モジュールと p-モジュールの重なりを調べる手数が計算の主要部であることを示している。従って、高速化のためには重なりを効率良く調べる方法が必要である。

さらに、146 モジュールを含む実際の PCB の製品データに $SA_{p-module}$ を適用してみた。面積の大きい順に 7 つのモジュールに対し、配置座標指定を行った。図 6.2 にその結果を示す。同データにおける配置座標の指定のない場合のパッキングが図 3.4 に示している。この結果より、 $SA_{p-module}$ は、座標指定のない場合と同等の高品質なパッキングを生成できることがわかる。

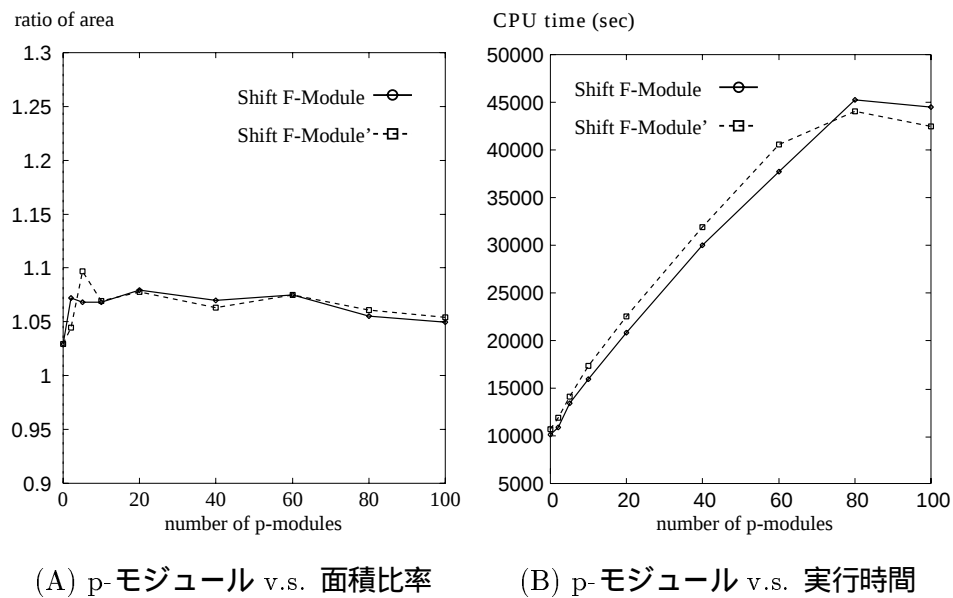


図 6.1: 200 モジュール

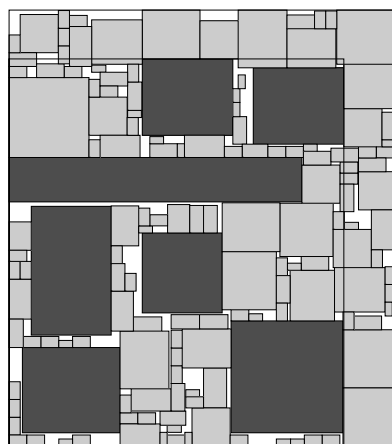


図 6.2: PCB の製品データ (モジュール数 : 146 , p-モジュール数 : 7)

6.3 レクトリニアモジュールを含むパッキング

$SA_{rectilinear}$ の性能評価のために、7 個のレクトリニアモジュールを含む 100 モジュールを入力データとして作成した。実問題においては、少数のレクトリニアモジュールと多数の矩形モジュールからなるデータが多いことを参考にしている。

このデータを $SA_{rectilinear}$ により、約 2 時間かけてパッキングした結果を図 6.3 に示す。この結果において、基板面積はモジュール面積の総和の 1.07 倍以下である。複雑なレクトリニアモジュールの入力に対して、従来にはみられない高品質なパッキング結果が生成されている。



図 6.3: 100 モジュールのパッキング (レクトリニアモジュール数 : 7 個)

第 7 章

おわりに

PCB 設計自動化の基本技術となる 2 次元パッキング問題において、従来のメタグリッド BSG による手法に自然な拡張を施し、設計者による配置座標指定のあるモジュールの取り扱いを可能とした。さらに、この座標指定のあるモジュールを扱う手法の拡張により、レクトリニアモジュールを 2 次元パッキングする手法の提案を行った。実験においては、実際の PCB の製品データを用いて、提案手法の性能の高さを示した。今後の問題としては、実用上は提案手法の高速化に尽きる。理論面では、提案手法では生成できない配置が存在するので、このことが実用上どのような障害になっているのかを確かめる研究が必要である。

謝辞

本研究を進めるにあたり，適切な御教示と絶えざる励ましを頂きました，東京工業大学 工学部 電気電子工学科 梶谷洋司教授ならびに，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，同大学 平石邦彦助教授に心から感謝致します。

また，有益な助言と多くの示唆に富んだ討議をして頂いた，東京農工大学 工学部 電子情報工学科 藤吉邦洋講師ならびに東京工業大学 工学部 電気電子工学科 高橋篤司助手に感謝致します。

数多の助言と貴重な時間を割き研究に御協力下さった，東京工業大学 工学部 電気電子工学科 中武繁寿助手に深謝致します。

様々な助言を下さった，北陸先端科学技術大学院大学 村田洋氏ならびに金子・平石研究室の諸氏，東京工業大学 高島康裕氏ならびに梶谷・上野研究室の皆様に感謝致します。

また，種々のご協力を戴いたアポロ技研株式会社システム部部長金子栄一様に深謝します。

参考文献

- [1] H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani, "Rectangle-Packing-Based Module Placement," in Proc. of ICCAD pp.472-479, 1995.
- [2] S.Nakatake, K.Fujiyoshi, H.Murata, and Y.Kajitani, "Module Placement on BSG-Structure and IC Layout Applications," in Proc. of ICCAD pp.484-491,' 1996.
- [3] M.Chi, "An Automatic Rectilinear Partitioning Procedure for Standard Cells," in Proc. of 24th DAC, pp50-55, 1987
- [4] M.Kang and W.Dai, "General Floorplanning with L-shaped, T-shaped and Soft Blocks Based on Bounded Slicing Grid Structure," in Proc. of ASP-DAC, pp265-270, 1997.
- [5] T.Lee, "A Bounded 2D Contour Searching Algorithm For Floorplan Design With Arbitrarily Shaped Rectilinear And Soft Modules," in Proc. of 30th DAC, pp525-530, 1993.
- [6] W.Dai, M.Sato, and E.Kuh, "A Dynamic and Efficient Representaiton of Bulding-Block Layout," in Proc. of 24th DAC, pp433-439, 1987.