

招待講演

## 配置の数理：多数の長方形を最小面積に埋め込む

梶谷洋司

東京工業大学 工学部 電気・電子工学科  
〒 152-8552 目黒区大岡山 2-12-1  
E-mail: kajitani@ss.titech.ac.jp

あらまし 多数の長方形を平面上に効率良く配置する問題が難しい理由を解析し、新しい技術の必然性を抽象し、それに応える一意四方位方式と呼ぶ方法を二通り紹介する。それぞれ BSG および Sequence-Pair と呼ばれるデータ構造に基づく。実装実験によれば数万個でも十分な品質で配置できるようになり、実用上限界は無くなったと言える。一般矩形、配線面積考慮、ソフト図形の配置、あるいは既配置図形、変形パッケージの扱い、更には3次元配置への発展、などの研究状況を紹介する。

キーワード： 配置, パッキング, クリティカルパス, BSG, Sequence-Pair

## On Packing of Rectangles into a Small Plane Area

Yoji KAJITANI

Dept. of Electrical and Electronic Engrg., Tokyo Institute of Technology  
Ookayama Meguro-ku, Tokyo, 152-8552 Japan  
E-mail: kajitani@ss.titech.ac.jp

**Abstract:** We discuss about possible reasons why the problem to pack rectangles on a plane into a small area is hard. Based on the conclusion, two new ideas, BSG and Sequence-Pair, are introduced which are special data structures, called the quad-direction-methods. Experiments show that they easily pack hundreds of rectangles by flat computaion, even tens of thousands rectangles if the method implemented hierachically. It is a breakthrough for the practical problem size. Several enhancements and developments are introduced. Examples are, packing of objects which are of general shape, routing area included, soft area constrained, positional constrained, and packing into an irregular area, and packing of cubes into a 3-dimensional space.

**Keywords :** placement, packing, critical-path, BSG, Sequence-Pair

# 1 はじめに

多数の平面図形(要素図形)を平面上に重ならないように置くことを配置といい、図 1.1 のイメージに示すように、全要素図形を囲む最小な長方形(領域)の面積の小ささが評価である。

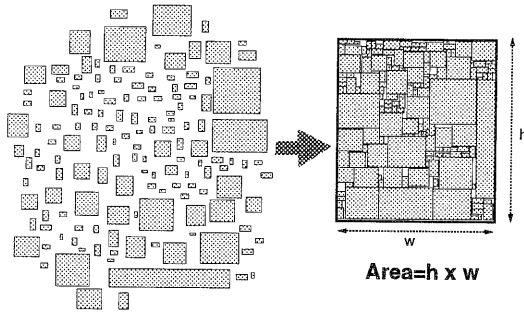


図 1.1 多数の要素長方形の長方形領域への配置

もし良い配置の方法があれば、駐車場の管理、部屋割、荷物積み込み、服地裁断、埋め込み型ゲームなど、様々な応用が考えられる。現在この期待が最も大きいのは、VLSI レイアウト設計と服地裁断(アパレル)の分野であろう。

ゲームでない限り、計算機援用(CAD)を考えるのは筋であるが、この目的に役立ちそうな計算機ができる処理は、

**衝突移動(1次元圧縮):** 水平(あるいは垂直)の直線を基準線として設定し、

1. 位置決め: 一つの要素図形を、基準線とも既配置図形とも重ならない位置におき、
2. クリティカル配置: 基準線に向かって、基準線あるいは既配置図形に重ならない限り、直線的に連続移動する

だけであるように見える。図 1.2 (A) には、基準線を水平線に決め、その上に図形を位置決めし、下向きに衝突移動させ(C)の配置を得る。

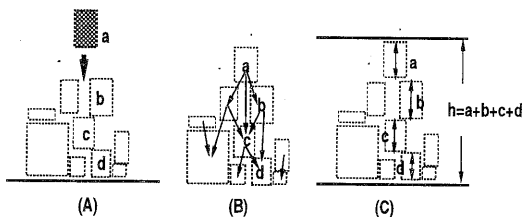


図 1.2 下向き衝突計算:  
上下制約グラフと高さクリティカルパス

従来から、非サイクル有向グラフ(DAG)の最長パス計算による配置方法、線形計画法による配置方法、スライス構造による配置方法などが知られているが、いづれも衝突移動を主計算としている。

衝突移動で得られる配置の高さ  $h$  は正確にいくつかの図形の高さの和になっている。それら図形の系列を垂直方向のクリティカルパスという。衝突移動を直交方向に繰り返し適用して配置する方法は、一次元配置法、あるいはクリティカルパス法と呼ばれる。すなわち、直交する2方向を決め、それぞれへの衝突計算を順に実行する。図 1.3 に示す例は、直交する2本の基準線を  $xy$  座標系に選び、要素図形を第1象限に置き、衝突計算を下向き続いて左向きに実行して(B)を得る。あるいは、逆順に実行して(C)を得る。この例では移動の向きの順によって結果が異なっている。

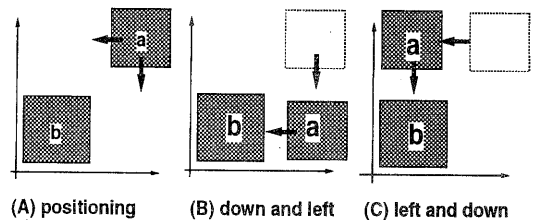


図 1.3 クリティカルパス法:

水平衝突, 垂直衝突の計算順による違い

クリティカルパス法は、現在の既配置物に対する水平と垂直の関係を挙げ、複数あるときには一方だけを取り上げそれを壊さないように限界まで(criticallyに)移動させる。図 1.3 を使って、移動の方向によって結果が異なる理由を解析する。要素図形  $b$  が既配置で  $a$  を衝突移動させようとしている。観察すると「(1)  $a$  は  $b$  の右にある」かつ「(2)  $a$  は  $b$  の上にある」。衝突移動をするにあたって(1)を保存する限り  $b$  と重ならない保証がある。そこで、 $a$  を下まで落してしまい、(B)の配置を得る。逆に、(2)を保証として衝突移動すると(C)の配置になる。いづれの場合も、「始めに観察された二つの関係のうち一方は破綻しても構わない」と衝突するのが衝突移動である。

クリティカルパス法には大きなメリットがある。すなわち、要素図形が凸形であれば、最小面積配置の一つを得ることができる。これを納得するには「正解をほどく」ことをして見れば良い。すなわち、一つの面積最小配置を考え、その右端に位置する(それより右に図形の無い)任意の要素図形を選び、他を動かさないように取り去る。以下再帰的に、現在の配置における左端の図形を、全てを尽くすまで、一つずつ取り去る。すると、明らかに、垂直に基準線を想定すればこの処理の逆が正解を得る衝突移動である。各要素図形の位置は必ずしも再現されないが、幅は変わらない配置を得る。

しかし、クリティカルパス法を配置アルゴリズムに仕立てあげるには大きい障害がある。すなわち：

**方向依存：**既配置図形に対し「上かつ右」のような合成方向にある図形の衝突移動は、多くの場合、他方を破綻させる。この場合、衝突の方向によって結果が異なる配置を得る。

**位置決め依存：**どこに図形を置いて衝突計算するかの多様性は、連続無限の多様性を生む。

上の2問題から目をそらす安易な方法は、問題の一般性を犠牲にすることである。例えば、直線構造に限定すれば、どの二つの図形も関係は左か右かが一意である。しかし、最小面積配置を求めるのが自明で問題にならない。スライス構造は、限定の強い配置しか出さないが、良いアイデアである。この場合、どの二つの図形も一意に左あるいは右の関係にあるから非一意性の問題は起きない。位置決めの多様性はシミュレーテッドアニーリング(SA)で、計算資源の問題に持ち込む方法[WL86]は良く知られ使われている。

衝突計算法に代わる鬼手があるかも知れない。しかし、ここでは「衝突計算法しか使わない」と自らを限定して、何を突破すればいいのかを再度真面目に考えよう。

2次元世界への配置であるから、方向依存は避けられない、ように思われる。始めに決めた関係を衝突移動後も保つのは実世界ではあり得ない。ここをブレイクスルーするには、抽象的世界の創造が必要である。

また、位置決めの多様性も避けられないように思われる。「中ほどまで詰める」ためには平面が離散化されていなければならないが、そうすれば最適配置を失う恐れがある。

ここに提案する二つの方法は両難問のうち方向依存性から解放された抽象世界を実現する。位置決めについては、有限通りの解空間を構成することにより、確率的探索法で逃げる戦略を採用する。こうして実用世界で通用する始めての配置自動化に成功するのである。

以下、2章では、良い解決策は一意四方位を実現する世界の構築にあることを述べる。3章では、その世界を実現する具体的な方法を紹介する。4章では、位置決めでSAで対処するとすれば、意味ある探索空間はP-propertyを満たさなければならない、と述べる。5章では、提唱する配置アルゴリズムを述べる。6章は提案方式の一般化で、7章は結言である。

なお、技術的な詳細は文献[NMFK96, NFMK98, MFNK96, MFNK95]を見られたい。

## 2 解法への道

衝突移動はどのような入力で動くかを確認したい。

図1.2の衝突移動では、「aはbの上、bはcの上、cはdの上」という情報から高さがわかり、他と比較してその値が最も長かったので、実行できた。すなわち、衝突移動は要素図形の対の間の上下関係が与えられれば可能である。ここに、衝突移動計算が、図(B)に示すような垂直関係を表すグラフ(枝は二つの図形が重ならないために離れなければならない最小距離を表す)の最長パスを発見せよ、というようにグラフアルゴリズムに帰着する理由がある。

話の一般性を誤解されないために、上下左右とか東西南北、前後左右のような直交する方向(の組)を一般に四方位と呼ぶことにする。

もともと、配置に衝突移動が使える理由は、それが「配置の条件」を守るぎりぎりの操作であるからである。配置の条件とは「どの図形対をとっても重ならない」ことである。そのため、衝突移動計算の入力の必要条件は：

**配置の条件：** どの二つの要素図形にも四方位が少なくとも一つ定義されている。

先の例で説明したように、実際の配置を観察すると、多くの場合、要素対は複数の四方位を満たしている。これは配置条件としては冗長である。これが引き起こす問題はそればかりではない。複合四方位は衝突移動の入力にはなり得ない。守られないスベックになるからである。つまり、構成アルゴリズムとしては入力は次の性質を満たさなければならない。

**一意四方位：** どの二つの要素図形にも四方位が丁度一つ定義されている。

この入力を実感するのは難しい。少なくとも実際の世界であり得ない。四方位は必ず合成四方位を生む：東西南北を定義すれば必ず「東北」地方が定義される。全平面のうち、丁度「東」の地方などは無視される。一意四方位の世界は抽象世界である。

とりあえず存在するとして、そのような世界では、先の例題で示唆したように、どちらの方向の衝突移動を実行しても、他方の四方位は(存在しないのであるから)破綻することはない。この論理から次に性質が導かれる。

**補題 2.1** 一意四方位を入力とすれば、衝突計算を水平方向、垂直方向いずれの順に処理するクリティカルパス法も同じ配置を導く。

よって、一意四方位を列挙することと配置を列挙することは1対1対応することになる。

すると、次の願望は、一意四方位を列挙する方法であるが、少々挙げても意味がない。とは言え「あらゆる配置」を列挙するのは、連続無限にあるから無理である。しかし「最適解をひとつでも含む配置集合」に

対応する一意四方位を列挙する方法を期待するのは、欲深いことではあるが、可能かも知れない。なぜなら、最小面積の多様性は有限であるからである：クリティカルパス法で求まる配置の全ての集合を $\Pi$ とすればこの中に正解が存在する。このことは先に述べた。図 1.2 で説明したように、クリティカルパス法で求まる配置の高さはクリティカルパスの長さであり、その多様性は高々要素図形の組合せの多様性、すなわち  $2^n$  通りしかない。幅についても同様である。したがって  $\Pi$  の要素数は高々  $(2^n)^2$  で有限である。

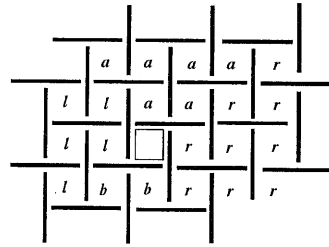


図 3.1 BSG と、その部屋間の四方位

### 3 一意四方位世界の実現

一意四方位を実現する最初のデータ構造は中武が発見した [NMFK94]。スライス構造のスライス線が全領域を切り分けてしまうのに対し、提案では線分が領域を一定の長さだけ切り開く。この差を強調して限界スライスライングリッド (bounded-sliceline grid, BSG) と命名した。これは抽象平面で、ここに要素図形を格納するだけで一つの一意四方位が生成されるのである。

BSG データ構造は 2 次元アレイである。この枠を取り去り、図形間に定義される関係だけを表すには図形の順列対で十分であることに思い至ったのは村田洋である [MNFK95]。この表現を Sequence-Pair (SP と略称する) と命名した。

BSG と SP とは、このように生い立ちは姉妹の関係にあり、数学的には互いに他をカバーできる関係にある。しかし、これから見ていくように、BSG はそのデータ構造が図式的であるゆえに、本来図式データである配置を直観的に把握でき、応用上に登場する制約を自然に採り込むことができる。一方、SP は、データ構造が置換であるため代数的な発想を誘う。

両データ構造の目的は共に「一意四方位を定義する」ことにあり、まとめて記述できる部分が多い。そこで両者を総称して一意四方位方式と呼ぶ。

#### 3.1 BSG による一意四方位生成

図 3.1 に示すような BSG-セグと呼ぶ 2 単位長の線分を 1 単位ずつずらして置かれた規則的なグリッドが BSG である。

区分けされた四角い空間を部屋と呼ぶ。任意の二つの部屋の間には一意に四方位が決まっている。図で□から見て  $a$  と記されている部屋は上 (above) にある。 $l$  は左 (left),  $b$  は下 (below),  $r$  は右 (right) で同様である。この決め方は、仮想的に□の上の壁を押しみると面積が縮まる部屋は上にある。そして、推移的に、上の上の部屋は上にある、との定義から決まる。このように決めれば、四方位のいずれか一つに決まっている部屋に全領域は分割される。

そして、要素図形を部屋に、1 部屋には高々 1 個、という規則で割り当てる。例として、図 3.2 に 4 図形  $a, b, c, d$  の割り当て例を示す。二つの要素図形間には、部屋の四方位を引き継がせる。このような過程を経て要素図形間の四方位が誘導される。

今の例では、

$$\begin{aligned} a & \text{ は } b \text{ の上} / c \text{ の上} / d \text{ の左} \\ b & \text{ は } a \text{ の下} / c \text{ の左} / d \text{ の左} \\ c & \text{ は } a \text{ の下} / b \text{ の右} / d \text{ の下} \\ d & \text{ は } a \text{ の右} / b \text{ の右} / c \text{ の上} \end{aligned} \quad (1)$$

補題 3.1 BSG への割当てから誘導される要素図形間の四方位は一意四方位である。

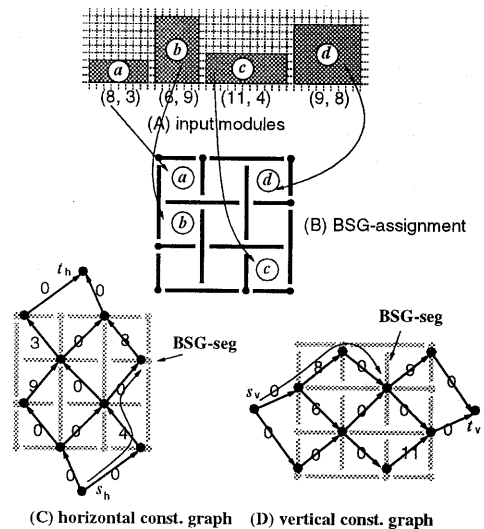


図 3.2 入力→割り当て→垂直、水平制約グラフ→クリティカル (最長) パス計算

水平垂直制約グラフを入力として衝突移動を計算すると、(要素図形の位置ではなく) BSG-セグの位置が決まる。そして、BSG の物理的サイズが決まる。各要素図形はその BSG の部屋の中に納まる。

今の例では、結果の配置図 3.3 では図形 d の左下の位置が  $(x = 8, y = 4)$  である。これを決める計算過程が図 3.2 に示してある。同図 (C) に示すように、d の下にある水平セグに対応する点に至る最長パス長は 4 である。同様に、d の左にある垂直セグまでの最長パス長は 8 に決まる。

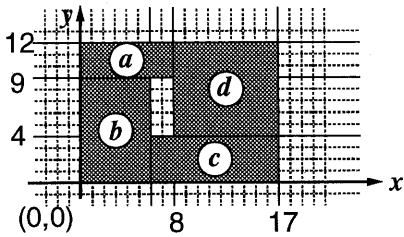


図 3.3 衝突移動による BSG-セグの位置

### 3.2 Sequenc-Pair による一意四方位生成

これは視覚を離れ、代数的に一意四方位を生成する。まず入力が 2 要素図形  $x, y$  の場合を説明する。両者の間の四方位は、一方を基準にとれば、4 通りでありこれらを次のように順列対で 1 対 1 に符号化する。

- $x$  は  $y$  の左  $\Leftrightarrow (xy, xy)$
- $x$  は  $y$  の右  $\Leftrightarrow (yx, yx)$
- $x$  は  $y$  の上  $\Leftrightarrow (xy, yx)$
- $x$  は  $y$  の下  $\Leftrightarrow (yx, xy)$

図 3.4  $x$  を基準にした四方位のすべて、と 2 文字順列対による符号化

2 つの要素図形のラベルの順列対が与えられればこの表を見て位相を解釈する。図形が 3 個以上あるときも順列対から関心のある図形対を抽出しこの表から四方対を解釈する。例えば、4 図形  $a, b, c, d$  のとき順列対  $(abdc, bcad)$  が与えられたとする。  $a, b$  を抜き出すと  $(ab \square \square, b \square a \square)$  であるから表から「 $a$  は  $b$  の上にある」と読める。このようにして、全図形の (名前の) 順列対が一つ与えられれば任意の図形対に対して一意に四方位を与えたことになる。なお、この順序対から得られる四方位は式 (1) に一致している。

補題 3.2 順序対から誘導される四方位は一意四方位である。

## 4 確率的探索法の効率条件

以上の話から、BSG の割り当て、あるいは SP を列挙することによって四方位を列挙し、衝突移動で配置を実現し、その中から良い評価を与える配置を選ぶ、

という筋書きが見えてくる。この時、列挙の仕方についてはいくつかの要件がある。

解空間確率的探索法は、計算機の速度向上により有効になって来た方法である。シミュレーテッドアニーリング法はそのひとつであるが、評価勾配法 (hill climb) に沿う直観的な工夫が可能であるし、また我々が採用して成功したと考えている方法なので、簡単に説明する。

何らかの方法で生成された有限個の配置の集合を解空間  $\Pi$  とする。  $\Pi$  に属す各々の配置  $\pi$  に対してそれを交換する演算 (ムーブ, move) を定義する。複数通りでも良い、その一つを  $\mu$  とし、変換結果を  $\mu(\pi)$  とする。

アルゴリズムの骨格は：

### SA 配置アルゴリズム

1. 一つの配置  $\pi \in \Pi$  を「現在の配置」とする。
2.  $\pi$  に対し、
  - (\*)  $\mu(\pi) \in \Pi$  であれば： $\mu(\pi)$  を評価し
    - ・評価が改善されれば  $\mu(\pi)$  を「現在の配置」に置き換える
    - ・そうでなくともスケジュールに従う確率で置き換える
  - (\*\*) もし  $\mu(\pi) \notin \Pi$  であれば：現在配置を保つ。
3. スケジュールに従って終了し、それまでの「現在配置」の中で最も評価の大きい配置を出力する。

□

できるだけランダムに解をサンプルし、その中で最も良い解を採用する単純な方法をランダムサーチと言う。良い SA への期待は

- 投入した計算資源に対し正解に収束する、かつ
- ランダムサーチより良い

ことである。それに応えるには、解空間  $\Pi$  とムーブ  $\mu$  は幾つかの性質を満たす必要がある。以下は最小の条件であろう。

### P-property

1. (正解 P-収束) 解空間には正解が含まれ、任意の配置から多項式数回のムーブで到着できる。
2. (P-評価) 配置  $\mu(\pi)$  の評価は多項式計算量で可能である。
3. (P-隣接) 各配置に対し、ムーブで変換される配置の多様性は多項式数である。
4. (解空間の解析性) 常に  $\mu(\pi) \in \Pi$  である。(SA 配置アルゴリズムで場合 (\*\*)) が発生しない。)

項1、2の必要性は自明である。項3は、もしある配置  $\pi$  に対しそれをあらゆる配置へ変換するムーブが定義されていれば、 $\pi$  が現在解になった場合、SAは、相互関係が無い配置を計算資源の限り探索する可能性がある。これはランダムサーチと同じになってしまう。項4については：SAは評価のムーブに対する連続性、すなわち「1回のムーブで変換される二つの配置は似ている。似ている配置は近い評価をもつ」を成立基盤としている。したがって、ムーブが解以外のものを生成するとこの性質が期待できない。

スライス構造方式に基づくSAは[WL86]で提案され広く使われているが、正解収束性が保証されていないのみならず、解析性も満たされていない。特に正解P-収束性のために、スライス構造の一般化についての研究が多数なされているが、段々複雑化し実用からは離れる一方である。一方、[OTT91]は、配置の全探索を目的として、図形対間の左右上下関係で配置を類別する方法を与えている。これは自然に、ムーブを「左右上下の置換」とする解空間を定義しSAによる探索的解法を可能にするアイデアを含んでいる。しかし、任意の置換が許されないので解析性を欠く。

以上のように、従来は構成的にも探索的にも不十分なままに推移していた。

## 5 一意四方位方式によるSA配置

二つの困難（方向依存、位置決め依存）のうち、方向依存は一意四方位を与えるBSGあるいはSPで解決する。しかし、位置決め依存は越えることができない。これにはただひたすらに、BSGでは割り当て、Sequence-Pairでは順列対、を取り替えることによって対処することになる。

### 5.1 BSGによる一意四方位配置

BSGは本来無限グリッドである。しかし要素図形を割り当てる領域を制限したい。領域が大きいほど幾何級数的に割り当ての多様性が増えるし、冗長性が増えよう。一方、余り狭いと最適解を見逃す危険がある。

水平一列に並んだときそのときに限り正解となる入力例が存在するから、割当を考えるBSGの領域は少なくとも  $n \times n$  の部屋数を持つ必要がある。次の事実の証明は簡単ではない。文献[MFNK96]などにあるgrid-dingと呼ばれる技法を用いて証明された[NFMK98]。

**定理 5.1** BSGの割当領域が  $n \times n$  の部屋を有すれば、解空間「あらゆる割当てが生成する一意四方位に対応する配置の全て」は「一対の部屋の内容の交換に対応する配置の変換」をムーブとして *P-property* を満たす。

### 5.2 SPによる一意四方位配置

図形ラベルの順列対に対するムーブはラベルの交換が最も自然である。

**定理 5.2** 解空間「あらゆる *Sequence-pair* が生成する配置の全て」は「順列対の一方の順列のラベル対の交換に対応する配置の変換」をムーブとして *P-property* を満たす。

図 5.1 は、SPによるSA配置の1例である。

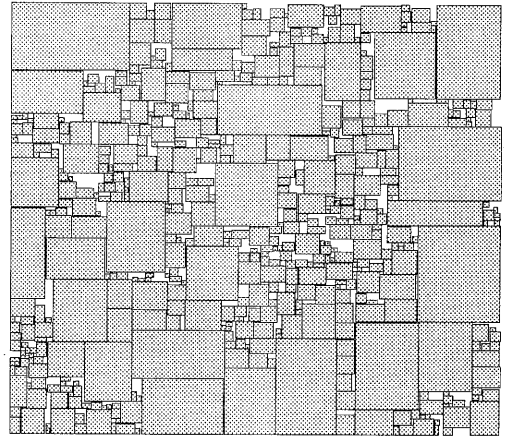


図 5.1 6時間SAによる500個の長方形の配置

$$\frac{\text{評価した配置数}}{\text{基板の面積}} \leq \frac{2,280,000}{(500!)^{2^{500}}} \sim \frac{1}{3 \times 10^{2412}}$$

## 6 一般化

今までは、要素図形が長方形で評価は長方形領域の面積という単純な定式化であった。応用に使うには実問題は余りにも複雑である。当初から、当然の一般化に加えて、諸応用それぞれにマッチするように工夫が加えられ発展がはかられている。

### 6.1 階層化

フラット計算でも千個程度の要素図形なら良好に配置できる。しかし、最適化アルゴリズムの入力規模限界を知るために階層化を試みた。すなわち、小さめのBSGを多数用意し、各BSGの部屋には、要素図形あるいは別のBSGを割り当てる。子に親が入らないように多分木の構造をとらせる。SAはこの木構造を取り替える。28,000個を十分な精度で配置した記録がある[WNK98]。

これだけの規模が扱えると新しい応用が湧いて来る。例えば、文献[SNK98]では、配置駆動の回路分割を提唱している。

## 6.2 一般矩形配置

VLSI レイアウト設計であれば、要素図形は最小機能単位である。性能を十分に確保して面積の小さい設計のためには図形の任意性を大きくしたい。様々な提案があるが、かなり広い一般手法を提案している文献 [SNK98] の一例を図 6.1 に示す。

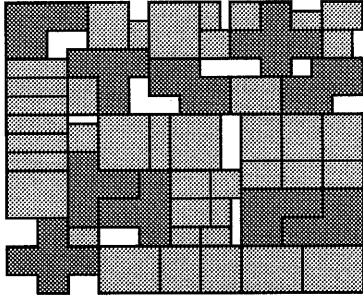


図 6.1 一般矩形の配置

## 6.3 制約つき図形配置

回路設計では、VLSI あるいはアナログ回路、あるいは PCB それぞれの設計スタイル毎の性能重視設計の帰結として

- 任意形状パッケージ
- ソフトモジュール (縦横比可変の長方形)
- 固定モジュール, 障害物
- 相対的な位置関係指定モジュールの組

などを扱う必要がある。これら制約つき自動配置を開発する必要は当初から問題意識にあり、多くの研究がある。文献 [FMMK97, MFK97, MFWK97, FM98, NFMK98] など参照されたいが、いずれも満足できる精巧さで自動化設計が可能になったと主張している。

## 6.4 配線領域こみの配置

モジュール相互の配線は、場合によっては全面積の半分を占める。配置を配線とは互いに他を決めなければ設計できない、という矛盾関係にあるので、それぞれの最適手法では限界がある。従来は再配置再配線の改善ループを計算資源の許す限り回す方法に走っている。しかし、もし配置と配線が同じデータ構造に格納されていればそれぞれ自動的に他方に影響するので単一問題になり、順序矛盾は消える。

それを可能にするのは、一つのデータ構造であらゆる場合を生成する (universal, 汎) データ構造でなければならない。BSG は配置の汎データ構造である。その

BSG-セグを配線チャンネルのモデルとして使えば配線に汎データ構造を得る。そして配置配線一体の汎データ構造となる。文献 [NSKK98] では、このアイデアを実装し実験では将来性のある結果を出している。

## 6.5 三次元立体配置

立体を詰め込む問題は、現在はコンテナ積み込みなどしか応用は考えられないが、人にとってイメージし難いだけに自動設計して見たい欲求がある。SP 方式であれば、pair を triple にすれば良い、ように思えるが、triple は 8 通りの組合せを生むのでそのままでは 4 次元パッキングになってしまう。従って制約つき sequence-triple による符号化を扱うことになるが、これで P-property 空間が作ることにについて研究を進めている。最適解への収束性についての保証はないが、とりあえず自動化は進んでおり、下図に結果の一例が示してある。

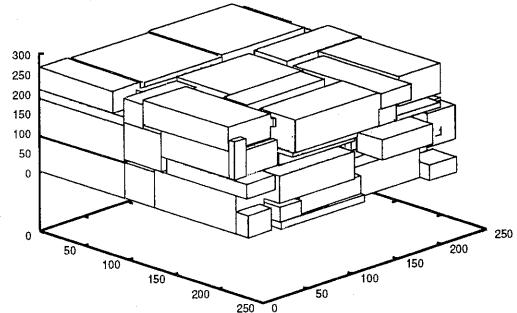


図 6.2 Sequence-Triple に基づく立体配置

## 7 おわりに

「衝突移動だけしか使えない」という境界条件での配置アルゴリズムは四方位方式に限られよう。他にもあり得るが、今までには BSG と SP が得られている。これが本文で伝えたかったことである。

我々は日米で特許化した [Patent 1~4]。しかし、完全に一般でありながら原理は従来からの唯一の自動化 (スライス構造) 方式よりずっと簡単で一読して実装できるよう。従って各応用問題毎に諸工夫ソフトの価値を決める。

この研究は 1992 年に開設された北陸先端科学技術大学院大学の一期生である博士課程生中武が得た 1994 年ころからの発想から始まり、同じ一期生村田と助手 (藤吉) を交えた討論を経て論文 [NFMK94] になった。その後、SP を得て実質的な意味を持つまでには日米の多くの大学人、企業人からの多大な助言や協力を頂いている。

特に村田は、社会人学生でもともと回路設計のエキスパートであるだけに、このアイデアの実際の有効性を直ちに理解し、カリフォルニア大 (Prof. Kuh at Berkeley) でベンチャーの風土を学び、日本で人材と資金を集め、ついに「マイクロアーク社」を本年6月に設立した。現在SP配置技法をベースにしたトップダウン配置配線レイアウトツールを開発中であり、性能を検証する実データ提供する協力を求めている。

本研究はいまも発展中であるが、今までの御協力を深謝し引き続き御尽力をお願いする次第である。

## 参考文献

- [WL86] D. F. Wong and C. L. Liu, "A new algorithm for floorplan area optimization," 23rd DAC., pp.101-107, 1986
- [OTT91] H. Onodera, Y. Taniguchi and K. Tamaru, "Branch-and-bound placement for building block layout," 28th DAC., pp.433-439, 1991
- [NMFK94] 中武, 村田洋, 藤吉, 梶谷, "モジュール配置問題を解く限定スライス構造の提案," 電子情報通信学会研究会, VLD94-66, pp. 19-24, 1994年10月
- [MNFK95] H. Murata, S. Nakatake, K. Fujiyoshi, and Y. Kajitani, "A solution space of size  $(n!)^2$  for optimal rectangle packings," 第8回回路とシステム軽井沢ワークショップ, pp. 109-114, 1995年5月
- [MFNK95] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, "Rectangle-packing-based module placement," ICCAD'95, pp.472-479, Nov. 1995.
- [NMFK96] S. Nakatake, H. Murata, K. Fujiyoshi and Y. Kajitani, "Module placement on BSG-structure and IC layout applications," IC-CAD'96, pp.484-491, Nov. 1996.
- [MFNK96] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair," IEEE Trans. on CAD Vol. 15, No. 12, pp. 1518-1524, Dec. 1996.
- [MFWK97] H. Murata, K. Fujiyoshi, T. Watanabe, and Y. Kajitani, "A mapping from sequence-pair to rectangular dissection," ASP-DAC, pp. 625-633, Jan. 1997.
- [FMMK97] 藤吉, 三輪, 村田, 金子, "ソフトモジュールを含む配置問題の一解法," 電情通学会 VLD96-104, pp. 63-70, 1997年3月.
- [SKTNK97] 坂主圭史, 倉澤剛, 高島康裕, 中武繁寿, 梶谷洋司: "BSG構造に基づく配置・概略配線同時最適化手法の提案" 電気情報通信学会技術研究会, VLD97, 登別6月(1997)
- [MFK97] H. Murata, K. Fujiyoshi, and M. Kaneko, "VLSI/PCB placement with obstacles based on sequence-pair", Intl. Symp. on Physical Design ISPD, pp. 26-31, April 1997.
- [MFK98] H. Murata, K. Fujiyoshi, and M. Kaneko, "VLSI/PCB placement with obstacles based on sequence pair," IEEE Trans. on CAD Vol. 17, No. 1, pp. 60-68, Jan. 1998.
- [FM98] 藤吉, 村田, "L型モジュールを含んだ矩形パッキング問題に対する sequence-pair を用いたアルゴリズム," 第11回回路とシステム(軽井沢)ワークショップ, pp. 113-118, 4月, 1998.
- [NFMK98] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module packing based on the BSG-structure and IC layout applications" IEEE Trans. on Computer Aided Design (採録決定)
- [NFK98] S. Nakatake, M. Furuya, and Y. Kajitani, "Module placement on BSG-structure with pre-placed modules and rectilinear modules," ASP-DAC, pp.571-576. (1998)
- [NSKK98] S. Nakatake, K. Sakanushi, Y. Kajitani, and M. Kawakita, "The channeled-BSG: A universal floorplan for simultaneous place/route with applications IC layout" ICCAD, 1998 (発表予定)
- [SNK98] K. Sakanushi, S. Nakatake, and Y. Kajitani, "The multi-BSG: Stochastic approach to an optimum packing of convex-rectilinear blocks" IC-CAD (1998) (発表予定)
- [WNK98] 呉, 中武, 梶谷 "高速ULSIフロアプランを実現するBSG階層化技術の提案" 情報処理学会DAシンポジウム, 浜松7月(1998)
- [Patent 1~4] Method of placing and extracting modules No.552,742, Sept. 29, 1997, 他1, 素材の配置および切り出し方法, 特開平9-108933, 1997年4月28日, 他1