社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－ＳＬＤＭ－112 (37)
2003／11／28

Selected Sequence-Pairを用いたレクトリニア多角形パッキングの高速化

池田 顕† 児玉 親亮† 中込 明広† 藤吉 邦洋†

† 東京農工大学 工学部 電気電子工学科, 〒184-8588 東京都小金井市中町 2-24-16
E-mail: {akira,kodamada,nakagomi}@fjlab.ei.tuat.ac.jp, fujiyosi@cc.tuat.ac.jp

あらまし VLSIレイアウト設計の自動化では、過去に設計されたレイアウト資産を再利用するため、垂直また水平線分だけからなるレクトリニア多角形（多角形）形状のモジュールを扱えることが望まれる。本稿では、部分矩形集合に分けられた多角形の配置を矩形配置表現であるselected sequence-pair (SSP) により表し、SSPから部分矩形数 (n) の線形時間で矩形パッキングを得る従来手法を用いることで、与えられたSSPに基づく多角形パッキングを $O((p+1)n)$ 時間で得る手法を提案する（p は単純な矩形を除く多角形の数）。そして計算機実験によりその有効性を示す。

キーワード Sequence-pair, Selected Sequence-pair, レクトリニア多角形, パッキング

A Fast Algorithm for Rectilinear Block Packing Using SSP

Akira IKEDA†, Chikaaki KODAMA†, Akihiro NAKAGOMI†, and Kunihiro FUJIYOSHI†

† Department of Electrical and Electronic Engineering, Tokyo University of Agriculture & Technology
2-24-16 Nakacho Koganei Tokyo, 184-8588, Japan
E-mail: {akira,kodamada,nakagomi}@fjlab.ei.tuat.ac.jp, fujiyosi@cc.tuat.ac.jp

Abstract In this paper, we propose a method to represent rectilinear block packing based on a selected sequence-pair (SSP). SSP can represent an arbitrary rectangle packing and it is decodable in linear time of its size. Here, each rectilinear block is partitioned into rectangle sub-blocks. Also, we propose an algorithm to obtain a rectilinear block packing in $O((p+1)n)$ time (n and p each is the number of rectangle sub-blocks and rectilinear blocks without simple rectangles) keeping all the constrains imposed by a given SSP. The proposed algorithm requires $O(n)$ time if p is constant. The effectiveness of the proposed method was confirmed by the experimental comparison.

Key words Sequence-pair, Selected Sequence-pair, rectilinear block, packing

1. はじめに

ナノミクロンプロセスにおけるVLSI設計開発では、回路規模の増大により多大な設計労力を必要とする一方で、設計期間の短縮と設計コストの削減が望まれている。このような要求はVLSI設計工程の1つであるレイアウト設計においても厳しく問われる。レイアウト設計では、多くの機能モジュールの配置配線を物理的に決定するため、VLSIの性能に大きな影響を及ぼす。高性能化かつ低コスト化のためには、期待の性能を発揮できるように配置配線するだけでなく、過去に設計された質の良いレイアウト資産の活用が望ましい。

VLSI内部では一般に機能モジュール単体は矩形であり、これらが複数組合わさることで、いくつかの機能が構成される。レイアウト資産はモジュールの集合であり、その配置形状は複数の矩形からなるレクトリニア多角形（外周が水平線分と垂直線分だけで囲まれた多角形、以降多角形）とみなすことが出来る。レイアウト設計工程の自動化では、このような多角形に対応する必要がある。

ところでレイアウト設計の自動化のために研究されてきた様々な問題の1つに、多数の矩形形状のモジュールを如何に密に配置するかという矩形パッキング問題がある。村田らが提案した矩形配置の表現方法であるsequence-pair (seq-pair) [2]は、どんな矩形パッキングでも表現可能であり、全てのseq-pairには対応する矩形パッキングが必ず存在するという特長がある。seq-pairは矩形パッキング問題を解くための画期的な手法であるが、多角形には対応していなかった。

その後、多角形を水平または垂直線分で分割して得られた部分矩形集合の配置をseq-pairにて表現し、与えられたseq-pairからその制約に基づく多角形パッキングを求めるために、矩形パッキングの場合と同様の制約グラフに多角形を復元するための枝を加える手法が提案された[7]。しかし $O(n^3)$ 時間（n は部分矩形数）を要し、とても遅かった。後に、枝を付加した制約グラフを簡単化することにより、多角形数を m とすると、より高速な $O(n^2+m^3)$ 時間で多角形パッキングを求める手法が提案された[5]。しかし、この手法でも m が大きいと遅かった。

そこで本稿では、制約グラフを用いた多角形パッキングが、矩形パッキングを行う部分と多角形を復元する部分からなることに着目する。前者では近年提案された矩形パッキング表現であるselected sequence-pair (SSP) [11]に基づいた、矩形数の線形時間でパッキングを得る手法を用い、後者では部分矩形数の線形時間で全ての多角形を復元する手法を用いることで、単純な矩形の数を除いた多角形数を p とすると、$O((p+1)n)$ 時間でSSP表現に基づく多角形パッキングを得る手法を提案する。そして、計算機実験により提案手法の有効性を示す。

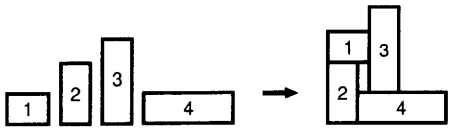
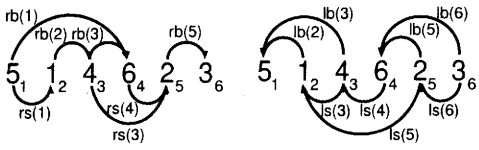


図 1　seq-pair (1234;2413) に対応する左下詰めパッキング



(a)rs, rb リンク　　(b)ls, lb リンク

図 2　標準化 $\Gamma_+=(514623)$ の各要素の rs, rb, ls, lb を求めた例

2. SSP と多角形パッキング

2.1 Sequence-Pair

sequence-pair [2] (seq-pair) では n 個の矩形の相対位置関係を、矩形名の順列 Γ_+ と Γ_- の対により、$(\Gamma_+;\Gamma_-)$ の形で表す。当然、n 個の矩形の配置について $(n!)^2$ 通りの表現がある。ここで、$\Gamma_+(i)$ は Γ_+ 中で第 i 番目の矩形を指し、$\Gamma_+^{-1}(a)$ は Γ_+ 中で左から矩形 a が何番目かを指す。Γ_- についても同様である。

seq-pair では矩形対の相対位置関係を、以下に示す「上下左右制約」として表す。Γ_+ と Γ_- で共に矩形 a が矩形 b の前にあるとき、つまり、$\Gamma_+^{-1}(a)<\Gamma_+^{-1}(b)$ かつ $\Gamma_-^{-1}(a)<\Gamma_-^{-1}(b)$ であるとき、矩形 a は矩形 b の左に位置する。また、Γ_+ では矩形 a が矩形 b の前にあり Γ_- では矩形 a が矩形 b の後ろにあるとき、すなわち $\Gamma_+^{-1}(a)<\Gamma_+^{-1}(b)$ かつ $\Gamma_-^{-1}(a)>\Gamma_-^{-1}(b)$ であるとき、矩形 a は矩形 b の上に位置する。例えば、seq-pair (1234;2413) は図 1 のような相対位置関係を表す。

2.2 Selected Sequence-Pair

Selected sequence-pair (SSP) [11] とは、要素数 n の seq-pair 中の隣接交差 [4] と呼ばれる部分列の数を、$n-\lfloor\sqrt{4n-1}\rfloor$ 以下に制限したものである。SSP は seq-pair と同様にどんな矩形パッキングも表現可能であり、どんな矩形パッキングにも対応する SSP が存在するという特徴がある。また、SSP は $O(n)$ 時間でパッキングにデコード可能であり、seq-pair ($O(n\log\log n)$ 時間 [9]) よりも高速である。

[11] では SSP の提唱と共に、要素数 n の SSP を矩形分割表現である Q-sequence (Q-seq) [10] に $O(n)$ 時間にて変換し、その Q-seq を矩形分割にデコードして各部屋に 1 対 1 対応する矩形を挿入することで、矩形パッキングを $O(n)$ 時間で得る手法が提案されている。しかしこの手法は Q-seq を介するため効率が悪かった。そこで本稿では、Q-seq を介さないで SSP から矩形パッキングを得る改良手法を提案する。本手法は時間複雑度が $O(n)$ 時間であり従来と変わらないが、Q-seq を介さない分効率的である。

2.2.1 SSP から矩形パッキングを求めるアルゴリズム

k 個の隣接交差を持ち、矩形 n 個の配置を表す SSP $\mathcal{S}$ から、以下に述べる 3 段階の処理を行うことによって $O(n+k)$ 時間にてパッキングを得ることができる。SSP は $k\leqq n-\lfloor\sqrt{4n-1}\rfloor$ なので、結果的に $O(n)$ 時間にてパッキングを得られることに注意されたい。

手続き SSP Decoder

Step 1: 与えられた SSP $\mathcal{S}$ 上の全ての隣接交差 (k 個) を列挙する ($O(n+k)$ 時間)。

Step 2: 先に得られた隣接交差の位置と列挙順に基づき、ダミー矩形を k 個挿入することにより隣接交差を全て除去する ($O(n+k)$ 時間)。ここで得られた隣接交差なし SSP $\mathcal{S}'$ のサイズは、ダミー矩形 k 個が加わったので $n+k$ になる。

Step 3-1: $\mathcal{S}'$ の各要素について、4 種の位置情報 rs, rb, ls, lb を求める ($O(n+k)$ 時間)。

Step 3-2: 先に得られた 4 種の位置情報と $\mathcal{S}'$ からアルゴリズム SSP–Packing によりパッキングを得る ($O(n+k)$ 時間)。

Step 1、**Step 2** 及び **Step 3-1** は既存手法 [11] である。本稿では、Q-seq を介さないでパッキングを得る **Step 3-2** の提案手法 SSP–Packing を中心に説明する。

説明の簡単化のため、一般性を失わず $\Gamma_-=(123\cdots n)$ となるように矩形名を付け替えたときの Γ_+ を標準化 Γ_+ と呼ぶ。例えば、SSP $\mathcal{S}'=(eadfbc;abcdef)$ なら、(514623;123456) と付け替えられ、標準化 Γ_+ は "514623" である。

Step 2 で得られた隣接交差のない SSP $\mathcal{S}'$ の標準化 Γ_+ を求める。そして **Step 3-1** において、この標準化 Γ_+ の各要素について以下の 4 種の位置情報を求める。

$\mathcal{S}'$ の標準化 Γ_+ の a 番目の矩形において、
a 番目の右で、$\Gamma_+(a)$ 未満で最も近い矩形の位置を $rs(a)$、
a 番目の右で、$\Gamma_+(a)$ より大きく最も近い矩形の位置を $rb(a)$、
a 番目の左で、$\Gamma_+(a)$ 未満で最も近い矩形の位置を $ls(a)$、
a 番目の左で、$\Gamma_+(a)$ より大きく最も近い矩形の位置を $lb(a)$。

標準化 Γ_+ "514623" の rs, rb を求めた例を図 2(a) に示す。同様に ls, lb を求めた例を図 2(b) に示す。

Step 3-2 では 4 種のリンクと標準化 Γ_+ から、提案アルゴリズム SSP–Packing によりパッキングを得る。SSP–Packing では、要素 $i-1$ が要素 i の右 (左) にあるとき、i から $i-1$ に向かって rs (ls) リンクを 1 回だけたどった後、$i-1$ に到達するまで rb (lb) リンクをたどり続ける。この一連のたどったリンクを ***rs-rb* (*ls-lb*) chain** と呼ぶことにする。

アルゴリズム　SSP–Packing

Input: $\mathcal{S}'$ の標準化 Γ_+ とその各要素の rs, rb, ls, lb (要素数 n)
Output: $\mathcal{S}'$ の示唆する制約に基づいた左下詰めパッキング
まず矩形 1 を xy 平面の原点に置く。
for($i=2,3,\cdots,n$){ /* 各部屋間の隣接関係チェック */
　if($\Gamma_+^{-1}(i-1)<\Gamma_+^{-1}(i)$){ /* 矩形 i は矩形 $i-1$ の右に配置 */
　　i から $i-1$ に向かって *ls-lb* chain をたどる。
　　矩形 $\Gamma_+(ls(i))$ の下辺の y 座標を Y、
　　ls-lb chain の全ての要素 (矩形) で
　　最も右側の右辺の x 座標を X とする。
　　矩形 i の左下角座標を (X,Y) とする。
　}
　else{ /* 矩形 i は $i-1$ の上に配置 */
　　i から $i-1$ に向かって *rs-rb* chain をたどる。
　　矩形 $\Gamma_+(rs(i))$ の左辺の x 座標を X、
　　rs-rb chain の全ての要素 (矩形) で
　　最も上側の上辺の y 座標を Y とする。
　　矩形 i の左下角座標を (X,Y) とする。
　}
}　　(アルゴリズム SSP–Packing 終.)

このアルゴリズムは、矩形分割へのデコードを想定しながら実行される。それゆえ、各矩形は分割線の位置を考慮しながら配置される。後述の提案多角形パッキングアルゴリズムでは

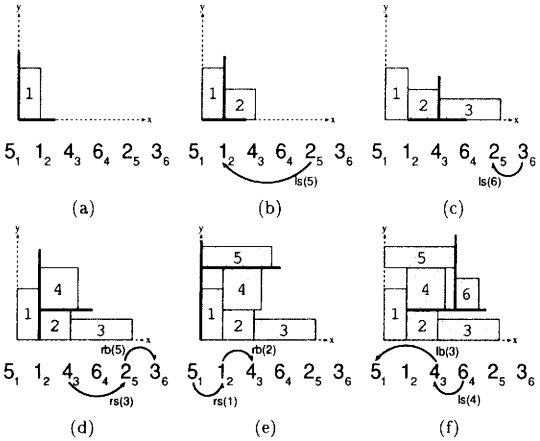
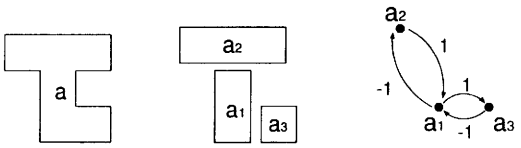


図 3 SSP $\mathcal{S}'=(514623;123456)$ に基づいた SSP-Packing によるパッキング例。太線は想定される水平または垂直分割線を表す。



(a) 多角形 a (b) 部分矩形集合 $\{a_1,a_2,a_3\}$ (c) 水平方向の特別枝

図 4 多角形の部分矩形集合への分割と特別枝

SSP のデコードを SSP から矩形分割を求める部分と、矩形分割からパッキングを求める部分とに分ける。矩形分割は部屋を枝、分割線を点とすることでグラフ表現可能であり、各点の水平（垂直）分割線の最長パスを求めることで、対応する部屋の左下 x (y) 座標を得ることができる [1]。すなわち、各部屋に入る矩形の左下 x (y) 座標を得ることができる。

アルゴリズムの実行例

入力を $\mathcal{S}'=(514623;123456)$ とする。まず、xy 平面の原点に矩形 1 の左下角がくるように配置する（図 3(a)）。続いて入力の標準化 Γ_+ の要素 2 に注目する。要素 1 が要素 2 の左側にあり、要素 2 から ls リンクをたどると要素 1 に到達するので、矩形 2 を矩形 1 の右に配置する（図 3(b)）。矩形 3 も同様に配置する（図 3(c)）。要素 4 は要素 3 の左側にあるので、要素 4 から rs-rb chain をたどり、矩形 4 を矩形 2 の左辺の右側かつ矩形 2 の上辺の上側に配置する（図 3(d)）。矩形 5 も同様に配置する（図 3(e)）。最後の要素 6 は要素 5 の右側にあるので、要素 6 から ls-lb chain をたどり、矩形 6 を矩形 5 の右辺の右側かつ矩形 4 の下辺の上側に配置する。図 3(f) では矩形 6 が浮いているが、これは $\mathcal{S}'$ の示唆する制約に基づいているのであり、誤りではない。

2.3 レクトリニア多角形パッキングの従来手法

外周が水平線分と垂直線分だけの多角形を**レクトリニア多角形**（以降、多角形）という。[7] では seq-pair を拡張して、凹凸を含んだ多角形パッキングを表現する手法が提案された。ここで、各多角形は水平／垂直線分で切断された図 4(b) のような部分矩形の集合として扱われる。

多角形パッキングにおいて、seq-pair から得られる制約グラフを用いた矩形パッキング手法では個々の多角形が復元されない。そこで seq-pair から得られた水平／垂直制約グラフに多角形を復元するための特別枝を付加する。いま多角形 a の部分矩形の名前を $a_1,a_2,a_3,\cdots$ とする。多角形 a の基準点から部分矩形 a_i の左下座標を $(X(a_i),Y(a_i))$ とし、切断された全ての部分矩形対（a_i,a_j とする）毎に、水平制約グラフに対し点 a_i から a_j に枝重み $X(a_j)-X(a_i)$ の有向枝と点 a_j から a_i に枝重み $X(a_i)-X(a_j)$ の有向枝を加える（図 4(c)）。同様に枝重みの計算に y 座標を用いた有向枝を垂直制約グラフに加える。そしてこの特別枝を付加した制約グラフから Ford の最短パスアルゴリズムを応用した最長パスアルゴリズムを用いて各点への最長パス値を求め、最長パス値を各部分矩形の左下座標とすることで左下詰めパッキングを得る。

このアルゴリズムでは $O(n^3)$ 時間で各点の最長パス値が求まるか、正の閉路を発見して終了する。正の閉路がある場合には対応するパッキングが存在しないので、このような seq-pair を**非許容な seq-pair** と言い、対応するパッキングが存在する seq-pair を**許容な seq-pair** と言うことにする。seq-pair が許容であるための必要十分条件は、特別枝を付加した制約グラフ上に正の閉路が存在しないことである。

3. 提案手法

3.1 提案手法での改良点

Ford のアルゴリズムを応用した従来の多角形パッキング手法は、特別枝を付加した制約グラフにおいて、もし点 v までの最長距離 $Dist(v)$ が点 u からの枝 (u,v) を通ることによって増加するなら、$Dist(v)$ を $Dist(u)+$ 枝 (u,v) の重み $Length(u,v)$ に変更する**緩和操作**を、全ての枝に対して行うことの繰り返しで構成されていた。ここで、枝を水平／垂直制約枝と特別枝に分けると、前者は ${}_nC_2$ 本以下、後者は $2n$ 本以下である（n は部分矩形数）。緩和操作は枝 1 本あたり定数時間なので、両者の全てにこれを行うと $O(n^2)$ 時間かかる。さらにこれら一連の操作を最大 n 回繰り返す必要があるため、全体では $O(n^3)$ 時間となり n が大きいと非常に遅いという欠点があった。

水平／垂直制約枝の緩和操作は、seq-pair が示す制約の下での矩形の左下詰め手法に、矩形を座標の増加方向にしか動かさないという制約を加えたもので置き換えることができる。そこでこの制約を満たすように変更可能な、SSP を用いた矩形の左下詰め手法に置き換える。

また、特別枝の緩和操作を、繰り返し毎に全ての多角形を復元するように変更する。これによりパッキングを得るために必要な全体の繰返し回数を最大 n 回から最大 $p+1$ 回に削減する。ただし、p は単純矩形の数を除いた多角形数である。

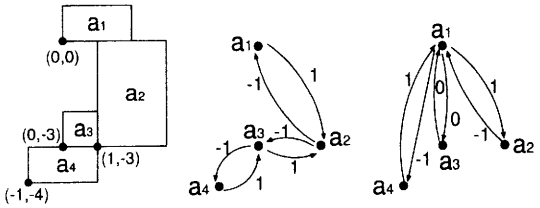
これらの改良により、$O((p+1)n)$ 時間で SSP の示す制約の下での多角形パッキングを得るか、その SSP は非許容であると判定する。

3.2 全ての多角形を復元する手法

本手法では、多角形毎に基準矩形を 1 個決め、全ての特別枝が基準矩形とそれが属する多角形の非基準矩形との間だけになるように張り替える。張り替えた枝の重みは、枝の始点から終点まで張り替え前の特別枝を辿ったときのパス長と同じにする。例えば図 5(b) の水平方向の特別枝は図 5(c) となる。垂直制約グラフも同様である。

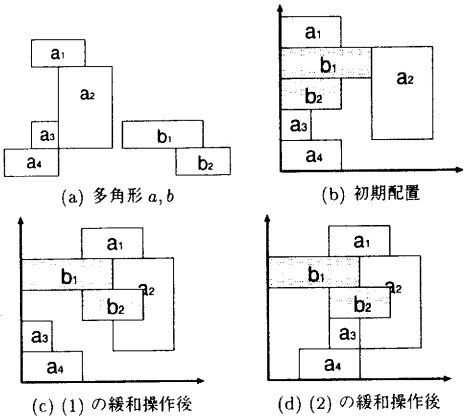
そして緩和操作を (1) 非基準矩形から基準矩形に向かう全ての特別枝、(2) 基準矩形から非基準矩形に向かう全ての特別枝、の順番で行う。これにより、繰り返しの度に部分矩形数の線形時間で全ての多角形が復元できる。

水平方向と垂直方向の特別枝は独立であるため、多角形を水平方向と垂直方向で独立に復元できる。散らばって配置された 2 個の多角形の x 方向での復元の様子を図 6 に示す。なお、この例では理解し易さのために y 方向は最終座標としてある。こ



(a) 多角形 a (基準矩形 a_1)　(b) 張り替え前の水平方向の特別枝　(c) 張り替え後の水平方向の特別枝

図 5　特別枝の張り替え



(a) 多角形 a, b　(b) 初期配置

(c) (1) の緩和操作後　(d) (2) の緩和操作後

図 6　x 方向の多角形復元 (基準矩形 a_1, b_2)

の手法では各多角形の復元のみを目的としているため図 6(c), (d) のような多角形同士の重なりは考慮しない。

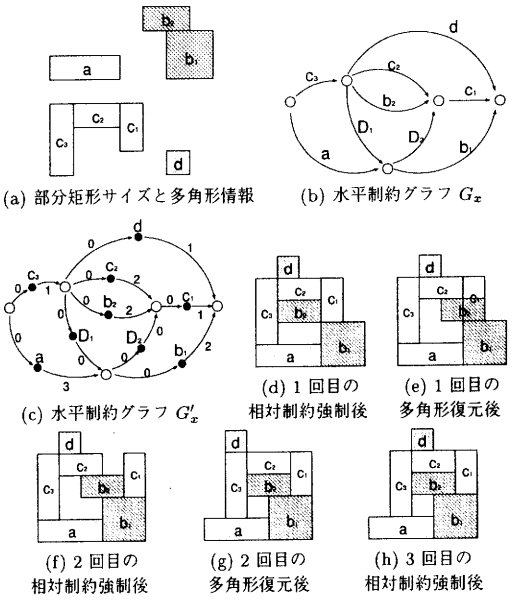
3.3 SSP に基づく多角形パッキングアルゴリズム

提案する多角形パッキング手法は、与えられた SSP に基づいた左下詰めパッキングを、水平方向と垂直方向で独立に計算して求める。以下に水平方向の計算をするアルゴリズム packingX を示すが、垂直方向の計算も同様である。なお、この関数は全ての部分矩形と垂直分割線の x 座標を返すが、収束しなかったときには非許容解であったと判定する。

```
packingX(SSP S', 矩形の幅 W, x 座標を表すベクトル x[ ],
                    多角形数 p, 水平方向の特別枝 E_x){
  S' から水平制約グラフ G_x を得る;
  G_x の各枝に、対応する矩形名の点を挿入し、
  そこまでの枝の重みは 0、そこからの枝の重みを対応する
  矩形の幅として水平制約グラフ G'_x を得る;
  ベクトル x[ ] の全要素を 0 とする;
  for (i=1, ··· ,p+1){
    相対制約強制 (G'_x, x[ ]);
    if (多角形復元 (E_x, x[ ]) == NOT_MOVE) return x[ ];
  }
  return "S' は非許容";
}
                              ( アルゴリズム packingX 終.)
```

ここで、全ての多角形を復元する "多角形復元 ()" は、部分矩形が移動しなかった場合に NOT_MOVE を返すとする。

3.1 節で述べた通りに水平／垂直制約枝の緩和操作を「SSP に基づいた矩形の左下詰め手法」に置き換えると、繰り返しの度に同一の SSP が同一の制約グラフに変換され無駄である。そこで packingX では高速化のため、SSP から制約グラフを得る変換は繰り返しの外に出してある。繰り返しの内側に残る部分では、SSP が示す水平制約を満たさなくなった部分矩形を、再び制約を満たす座標に移動させるために、制約グラフの各点への最長パス値を求めて部分矩形や分割線の座標を決定している。この操作を**相対制約強制**という。



(a) 部分矩形サイズと多角形情報　(b) 水平制約グラフ G_x

(c) 水平制約グラフ G'_x　(d) 1 回目の相対制約強制後　(e) 1 回目の多角形復元後

(f) 2 回目の相対制約強制後　(g) 2 回目の多角形復元後　(h) 3 回目の相対制約強制後

図 7　水平方向のパッキングの実行例

相対制約強制は $O(n)$ 時間、多角形復元手法は $O(n)$ 時間、最大 $p+1$ 回の繰り返しなので、全体では $O((p+1)n)$ 時間にて、SSP の制約に基づく多角形パッキングを得るか、その SSP は非許容だと判定することができる。

例として、図 7(a) に示した多角形 b, c $(p=2)$ と単純矩形 a, d が、SSP $S=(c_3\,d\,c_2\,b_2\,a\,c_1\,b_1; a\,c_3\,b_1\,b_2\,c_2\,c_1\,d)$ に基づいてパッキングされる過程を示す。まず、S は隣接交差を 2 つ含むため、ダミー矩形 D_1, D_2 が挿入されて隣接交差のない SSP $S'=(c_3\,d\,c_2\,b_2\,D_1\,a\,D_2\,c_1\,b_1; a\,c_3\,D_1\,b_1\,D_2\,b_2\,c_2\,c_1\,d)$ が得られる。次に、x 座標を求めるために S' から図 7(b) に示した水平制約グラフ G_x を得る。これを変更することにより、図 7(c) に示した水平制約グラフ G'_x を得る。ここで、黒点は付記した名前の部分矩形に対応し、白点は垂直分割線に対応する。その後、各部分矩形の x 座標が決められていく過程を図 7(d)–(h) に示すが、ここでは理解し易さのため y 座標を決定済として図示してある。多角形復元の後では部分矩形が重なっている場合があり、最後を除く相対制約強制後では復元できていない多角形がある。

提案アルゴリズムの正当性は、以下の定理により示される。

【定理 1】　提案アルゴリズムは、与えられた SSP が許容であるなら左下詰めパッキングを得ることができ、非許容であるならこれを判別することができる。但し、n は部分矩形数で、p は単純矩形を含まない多角形数である。

証明：　もし SSP が許容であったなら、定義から、その SSP に基づいた左下詰めパッキングが存在する。そこでの各部分矩形の座標を『最終座標』と呼ぶものとする。すると、[7] にて示されているように、各部分矩形の x/y 最終座標は、多角形に対応させるために特別枝を付加した水平／垂直制約グラフ上での対応する点への最長パス長と等しい。

一般性を失うことなく、3.2 で触れたのと同じに、制約グラ

フ上の全ての特別枝は多角形毎に1つの基準矩形を端点とする。また、制約グラフ上の点について、単純矩形に対応する点を『単純矩形点』、単純矩形を除いた多角形の基準矩形に対応する点を『基準矩形点』と呼ぶことにする。以降水平方向だけについて示すが、垂直方向についても同様である。

水平制約グラフにおいて、点毎にそこに至る最長パスの中で通過する基準矩形点の数が最も少ないパスに注目し、これが通過している基準矩形点の数を『最長経路最小基準矩形通過数(MNRVLP)』と呼ぶことにする。但し、非基準矩形に対応する点については、その基準矩形に対応する基準矩形点は数えないこととする。パスはその定義から、1つの点を1回しか通ることができないので、MNRVLPはp以下であり、単純矩形点以外の点のMNRVLPは$p-1$以下である。

1回目の相対制約強制の終了後、MNRVLPが0である単純矩形点は必ず最終x座標に至っていることが容易に分かるであろう。そして、1回目の多角形復元の終了後、MNRVLPが0である全ての点は必ず最終x座標に至っていることは容易に分かるであろう。

もし、MNRVLPがk以下である点が全て最終x座標に至っているなら、これに対して相対制約強制を1回施すと、MNRVLPが$k+1$である単純矩形点は必ず最終x座標に至る。また更に多角形復元を1回施すと、MNRVLPが$k+1$である全ての点は必ず最終x座標に至る。

従って帰納法により、p回目の多角形復元の終了後、MNRVLPが$p-1$以下である全ての点は必ず最終x座標に至っている。そして、$p+1$回目の相対制約強制の終了後、MNRVLPがp以下である単純矩形点は必ず最終x座標に至っている。故に、p回目の相対制約強制の終了後、入力が許容SSPであれば、全ての点の最終x座標が必ず求まる。

多角形復元において緩和操作が行われず、どの点のx座標も変化しなかったなら、多角形復元の入力において既に全ての多角形が原型を保てるx座標だと保証でき、その前の相対制約強制において相対位置の制約を満たしているので、これにより左詰めパッキングが完了したことが判定できる。相対制約強制についても同様であるが、1回目の相対制約強制だけは直前に多角形復元が行なわれていないため、例えどの部分矩形も動かなかったとしても、最終x座標に至ったとは言えない。

そして、p回目の多角形復元でもまだ緩和操作が行われるのであれば、入力が非許容SSPであると判定できる。

4. 計算機実験

4.1 デコード（表現から対応するパッキングへの変換）時間の比較実験

デコード時間を従来手法と比較するため、提案手法を[13]の隣接SSP解生成手法に組み込んでC言語にて実装した。また、比較相手として[5]の手法(SVX)をC++にて実装した。SVXは1つのseq-pairから対応するパッキングへのデコードを$O(n^2+m^3)$時間（nは部分矩形数、mは単純矩形を含む多角形数）で行うことができる。

そして、多角形が横一列に並んだSSPもしくはseq-pairを初期解として、これから隣接解を作成し許容なら現在の解と置き換えるという操作を、許容解と非許容解がともに10000個以上得られるまで繰り返し、表現1つ当たりのデコード時間の平均を許容解と非許容解に分けて測定した。

3種の入力、(1) $n/4$個の部分矩形からなる多角形が1個と、単純矩形が$(3/4)n$個。すなわち、$p=1$（pは単純矩形の数を除いた多角形数）、$m=(3/4)n+1$。(2) $n/16$個の部分矩形からなる多角形が16個。すなわち、$p=m=16$。(3) 2個の部分矩形からなる多角形$n/2$個。すなわち、$p=m=n/2$、の各々についてnが$32,64,\cdots,8192$のときにPentium4 2.4GHzの計算機で実験した結果を図8,9,10にそれぞれ示す。図から分かるように、全てについて提案手法の方がSVXより高速であった。

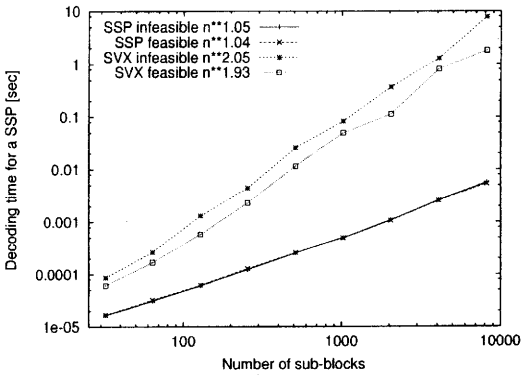


図8 提案手法(SSP)と従来手法(SVX)との表現1つ当たりのデコード時間比較（多角形数$p=1$、$m=3n/4+1$）

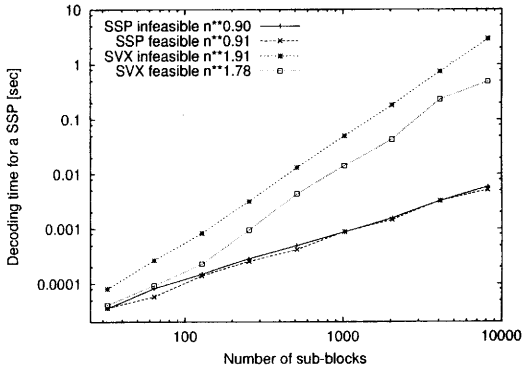


図9 提案手法(SSP)と従来手法(SVX)との表現1つ当たりのデコード時間比較（多角形数$p=m=16$）

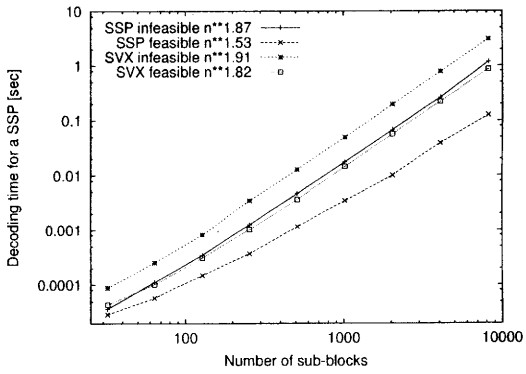


図10 提案手法(SSP)と従来手法(SVX)との表現1つ当たりのデコード時間比較（多角形数$p=m=n/2$）

(1)では$p=1$、$m=(3/4)n+1$なので提案手法は$O(n)$、SVXは$O(n^3)$、(2)では$p=m=16$なので提案手法は$O(n)$、SVXは$O(n^2)$、(3)では$p=m=n/2$なので提案手法は$O(n^2)$、SVXは$O(n^3)$であるが、それぞれの傾きを最小二乗法により計算した結果$O(n)$以外のものの多くはこれより傾きが小さ

表 1　SA 法による解探索結果の比較

手法	対象	計算機	時間 [s]	面積率 [%]
提案手法	多角形	AthlonXP 1700+	80	106.3
[7]	多角形	Ultra SPARC 200MHz	147	106.7
MPBSG [14]	コア凸多角形	Pentium III 910MHz	374	108.6
[8]	L 型	Celeron 400MHz	11	109.3
[6]	L 型	Ultra SPARC 200MHz	297	109.8
BSG [3] [12]	L 型	SS5 80MHz	300	111.5

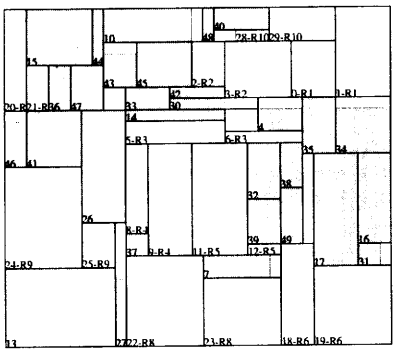


図 11　提案手法を用いた SA 法による解探索の結果
（L 型ブロック数 10、単純矩形数 30）

かった。この理由としては、O 記法は最悪の場合を想定しており $O(n)$ 以外の場合では平均的にはより高速に収束したことや、O 記法はサイズが十分大きい場合についての理論であり、この場合はサイズがまだ十分大きくないことなどが考えられる。

4.2　SA 法による解探索

本手法を SA 法に組み込み、解の探索を行った。図 11 は、[3] などで用いられたのと同じ問題（L 型ブロック 10 個、矩形ブロック 30 個）に対し、回転や反転を考慮せずに AthlonXP 1700+の計算機で 80 秒かけ探索した結果である。色が薄いものは、Rの後の番号が同じものとL型ブロックを構成している。同じ問題に対する SSP と従来手法との探索結果の比較を表 1 に示す。なお面積率とは、探索結果の外周矩形の面積をブロックの全面積で割ったものである。

更に、[7] の Fig.12(c) と同じ問題（多角形ブロック 14 個、矩形ブロック 5 個）を用いた解探索の結果を図 12 に示す。[7] では Ultra SPARC 200MHz の計算機で 13446 秒探索し外周矩形の面積が 468 であるのに対し、提案手法は AthlonXP 1700+ の計算機で 2251 秒探索し面積は 441 という結果が得られた。

これらの結果から、従来手法と比べて良い解が得られることが確かめられた。

5.　ま　と　め

レクトリニア多角形パッキング問題に対し、SSP を用いた矩形パッキング手法を用いる等の改良により、部分矩形数を n、単純な矩形を除いた多角形の数を p として、1 つの SSP に基づく多角形パッキングを $O((p+1)n)$ 時間で得る手法を提案した。さらに、実験により実際の計算時間を測定し、従来手法に比べ提案手法が平均的にも高速であることを確認した。特にパッキング対象の多角形数が固定である場合には、矩形数の線形時間で多角形パッキングを得ることができ非常に高速であった。また、提案手法を SA 法に組み込み、従来手法と比べて良い解が得られることを確認した。

謝辞　本研究は CAD21 プロジェクトの一部である。

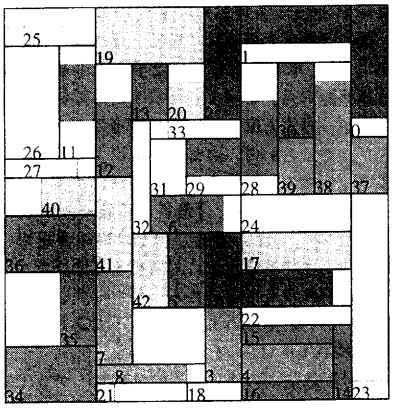


図 12　提案手法を用いた SA 法による解探索の結果
（多角形数 14、単純矩形数 5）

文　献

[1] Sediq M. Sait and Habib Youssef: "VLSI Physical Design Automation," IEEE Press, 1995

[2] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani: "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair," IEEE Trans. CAD, vol.15, no.12, pp.1518-1524, 1996.

[3] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani: "Module Placement on BSG-Structure and IC Layout Applications," in Proc. ICCAD'96, pp.484–491, 1996.

[4] 渡辺知巳, 村田洋, 藤吉邦洋: "Sequence-Pair で表されるモジュールの相対位置を保つ最適フロアプラン," 第 9 回 回路とシステム (軽井沢) ワークショップ, pp.457-462, 1996.

[5] 町田憲, 藤吉邦洋: "Sequence-Pair を用いたレクトリニア多角形パッキングの効率化," 信学技報 VLD2000-134, pp.1–6, 2001.

[6] 藤吉邦洋, 村田洋: "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, 1998.

[7] K. Fujiyoshi and H. Murata: "Arbitrary Convex and Concave Rectilinear Block Packing using Sequence-Pair," IEEE Trans. CAD, vol.19, no.2, pp.224-233, 2000.

[8] 齋藤宏明, 藤吉邦洋: "L 型ブロックパッキングの高速化に関する研究," 第 13 回 回路とシステム (軽井沢) ワークショップ, pp.245–250, 2000.

[9] X. Tang, R. Tian, and D.F. Wong: "Fast evaluation of sequence pair in block placement by longest common subsequence computation," IEEE Trans. CAD, vol.20, no.12, pp.1406–1413, 2001.

[10] K. Sakanushi, Y. Kajitani and D. P. Mehta: "The quarter-state sequence floorplan representation," IEEE Trans. CAS-I, vol. 50, no. 3, pp.376–386, 2003.

[11] C. Kodama, K. Fujiyoshi, "Selected Sequence-Pair: An Efficient Decodable Packing Representation in Linear Time using Sequence-Pair," in Proc. IEEE ASP-DAC, pp.331–337, 2003.

[12] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani: "Module Packing Based on the BSG-Structure and IC Layout Applications," IEEE Trans. CAD, vol.17, no.6, pp.519–530, 1998.

[13] 藤吉邦洋, 児玉親亮, 清田紘司: "Selected Sequence-Pair のための効率的な隣接解生成手法," 信学技報 VLD2002-6, pp.31–36, 2002.

[14] 坂主圭史, 中武繁寿, 梶谷洋司: "多層パラメトリック BSG によるコアセルの配置アルゴリズム," 信学論 (A), vol.J85-A, no.9, pp.938–949, 2002.