ARBITRARY CONVEX AND CONCAVE RECTILINEAR BLOCK PACKING USING SEQUENCE-PAIR

Kunihiro Fujiyoshi [1] *Hiroshi Murata* [2]

[1] Department of Electrical and Electronic Engineering
Tokyo University of Agriculture & Technology, JAPAN
fujiyosi@cc.tuat.ac.jp
[2] Micro Ark, JAPAN
murata@jaist.ac.jp

ABSTRACT

The sequence-pair was proposed in 1994 as a representation of the packing of rectangles of general structure. Since then, there have been efforts to expand its applicability over simple rectangles. This paper proposes a new way to represent the packing of a set of rectilinear blocks, including arbitrary concave rectilinear blocks. Our idea is in representation of a general block by a collection of rectangle blocks with additional constraints. Some sequence-pairs of rectangle blocks with such constraints may not be feasible, i.e. there is no corresponding packing. A necessary and sufficient condition of feasible sequence-pair is given by the properties of the horizontal and vertical constraint graphs. Furthermore, it is proved that any packing is represented by a feasible sequence-pair. The condition includes dimensions of blocks involved. However, for L-shaped blocks, an idea is given to represent them only in terms of the topology of the sequence-pair. A packing algorithm is designed as an SA search of the generated sequence-pairs. Experimental results show effectiveness of the proposed method.

1. INTRODUCTION

Recent advance of sub-micron technology makes it possible to realize a big system on a single chip. Designing such a huge VLSI layout is hard and so design reuse has been attracting much interest. In the layout design of such VLSI systems, blocks are not often simple rectangles. Therefore packing algorithm for arbitrary shaped rectilinear blocks is hoped. The following problem is the core of the practical requirements.

Rectilinear block packing (RP) problem: Let $\mathcal{M}$ be a given set of arbitrary rectilinear blocks. Pack $\mathcal{M}$ in the smallest rectangular area. Here, packing of a set of blocks is to place all the blocks without overlapping each other. □

Rectilinear blocks can be classified into two types of blocks: **convex rectilinear blocks** and **concave rectilinear blocks**. A rectilinear block is said to be **convex** if any two points in the block have a shortest Manhattan path inside the block. Otherwise, the block is said to be **concave**.

If $\mathcal{M}$ is consisting only of rectangles, RP is called **rectangular-RP**. The sequence-pair (seq-pair) [5, 7] and BSG [6, 16] were proposed to represent rectangular-RP of general structure. It was shown that they have properties preferable for generating a solution space and search. As for the seq-pair, every seq-pair is feasible, i.e. there is a packing corresponding to any seq-pair. There is a seq-pair which corresponds to the optimum packing. From an seq-pair, the corresponding packing is obtained in $O(n^2)$ where $n = |\mathcal{M}|$. Their practicality has been proved in stochastic search by experiments.

Intending to enhance the seq-pair and BSG to be able to handle rectilinear blocks such as L-shaped, convex, and concave ones, several ideas have been proposed. Their efforts have been to represent the RP packings for generating a solution space and search, by seq-pair [9], [13], [15], [17] and by BSG [6, 16], [8], [12], [14], [18]. The only method that can pack concave rectilinear blocks is [12]. However, [12] and [6, 16], [8], [9], [13] cannot represent a certain class of packings.

This paper gives a solution in the following fashion. A new method for RP is proposed based on the seq-pair which includes (1) to represent RP by seq-pair, (2) to give a proof that any RP is represented by seq-pair, (3) to realize a packing for a feasible seq-pair in cubic order, and (4) to give a necessary and sufficient condition for a feasible seq-pair.

The condition for (4) includes the dimensions of blocks. However, it is shown that if the blocks are L-shaped and rectangular, the condition is simplified so as not to use dimensions of blocks.

2. PRELIMINARY

2.1. Sequence-pair

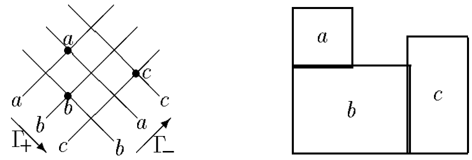
A sequence-pair [5, 7] for a set of n rectangular blocks is a pair of permutations of the n block names. For example, $(abc; bac)$ is a seq-pair for block set $\{a, b, c\}$. It is easily understood that the variety of the seq-pair for n blocks is $(n!)^2$.

A seq-pair imposes a **horizontal/vertical (H/V) constraint** for every pair of blocks as follows:

$$(\cdots a \cdots b \cdots ; \cdots a \cdots b \cdots) \quad \rightarrow \quad \left\{ \begin{array}{r} \text{the right side of } a \text{ is} \\ \text{left of the left side of } b \\ (a \text{ is left of } b) \end{array} \right.$$

$$(\cdots b \cdots a \cdots ; \cdots a \cdots b \cdots) \quad \rightarrow \quad \left\{ \begin{array}{r} \text{the upper side of } a \text{ is} \\ \text{below the lower side of } b \\ (a \text{ is below } b) \end{array} \right.$$

For example, sequence pair $(abc; bac)$ imposes a set of H/V constraints: $\{a$ is left of c, b is left of c, b is below $a\}$.



(a) Oblique grid (b) Packing

Figure 1. Example of oblique-grid and packing of seq-pair $(abc; bac)$

The H/V constraints of a seq-pair can be intuitively grasped using the **oblique-grid** notation. For example, Fig.1(a) shows the oblique grid of seq-pair $(abc; bac)$. It is an $n \times n$ grid obliquely drawn on the plane which is constructed so that the first sequence is observed along the sequence of the positive slope lines from left to right and the second sequence is observed similarly with respect to the negative slope lines. It shows the H/V constraints: block c is in the right quarter view range (between $-45°$ and $+45°$) of block a on the oblique grid, then c should be placed to the right of a.

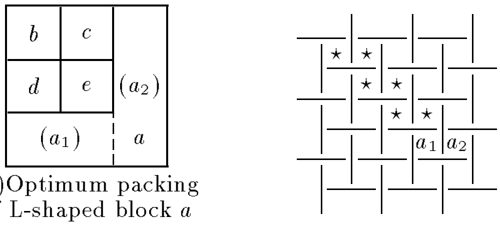
It has been proved in [5] and [7] that the set of H/V constraints imposed by every seq-pair is feasible, and an area minimum packing under the constraint can be obtained in polynomial time, and further, there is a seq-pair which leads an (globally) area minimum packing. Then, the seq-pair is easily utilized as a coding scheme of a stochastic algorithm.

To construct an area minimum placement for a seq-pair, 1-D compaction is carried out under the H/V constraints of the seq-pair. The blocks are greedily pushed to the left and to the bottom as shown in Fig.1(b). The resultant placement is called the **realization** of the seq-pair.

The realization can be obtained in $O(n^2)$ time by using the **H/V constraint graph** which is constructed faithfully to the H/V constraints. More in detail, Step 1 constructs a vertex weighted directed acyclic graph whose vertex set corresponds to the blocks and whose edge set corresponds to the horizontal constraints in the direction from left to right. The weight of each vertex is the width of the corresponding block. Determine the X coordinate of each block by the longest path length from the source node to the node of the block. Step 2 determines the Y coordinate of each block in a similar way using the vertical constraints in the direction from bottom to top.

2.2. Previous researches for RP problem

In [6, 16], a method to pack L-shaped blocks based on BSG is proposed: Partition each L-shaped block into two rectangles and place them onto two adjacent BSG-room. As two adjacent BSG-room have a common BSG-line, these two rectangles can be aligned by post-process easily. However, this method has a limitation that it cannot represent a certain L-shaped block packing. So, the optimum packing may not be represented by this method. For example, a packing shown in Fig.2(a) cannot be represented. As rectangles b, c, d, e are above a_1 and left of a_2, they must place onto BSG-rooms which are located upper-left direction. (BSG-rooms marked with $\star$ in Fig.2(b)) But in these limited BSG-rooms, relative positions of b, c, d, e in Fig.2(a) cannot be represented. After all, relative positions shown in Fig.2(a) cannot be represented by this method using BSG. (Using seq-pair based method proposed in this paper, relative positions shown in Fig.2(a) can be represented easily, for example, by $(bcdea_1a_2; a_1debca_2)$.)



(a)Optimum packing of L-shaped block a and rectangles b, c, d, e.

(b)BSG-rooms, where Rectangles b, c, d, e must be placed onto the room marked with $\star$

Figure 2. Example where BSG cannot represent the optimum packing

A more complicated method to pack L-shaped and T-shaped blocks using BSG is proposed in [8], but by the same reason, a certain packing cannot be represented. In [9], a seq-pair based method using a complicated relationship to pack rectilinear blocks is proposed, but overlaps may occur.

In [12], a BSG based method to pack rectilinear blocks allowing concave blocks and preplaced blocks are proposed. But this method may not represent the optimum packing as the authors point out.

In [13], a seq-pair based method to pack rectilinear blocks are proposed. Rectilinear blocks are partitioned into rectangles and post-process aligns rectangles to rectilinear block after the rectangle packing. This method can only pack mountain-shaped rectilinear blocks.

In [14], a topology based convex rectilinear block packing using BSG structure is proposed. This method is very complicated and a topology of initial packing cannot be represented and packed by the same reason with [6, 16]. ("Topological compaction" can be done easily, by using the proposed method in this paper.)

In [17], a method based on the seq-pair that can represent any convex rectangle packings is proposed.

In [18], a BSG based method which treats each convex rectilinear block as a sum of some rectangles is proposed. This method is proved to be able to represent any rectilinear packings. But multiple-BSG needs very big code space.

2.3. 1-D compaction using constraint graph

A useful method to compact the block placement in one dimension uses constraint graph [1, 3]. The H-constraint graph is made from initial positions of blocks. The set of vertices of the constraint graph consists of one source vertex and the others that correspond to the blocks. Edge (a, b) whose weight is equal to the width of a is introduced if block b is right of block a. Clearly, the weight of an edge is positive. Positive edge (a, b) of weight x means "b should be right of a by x or more than x". This constraint graph is a directed acyclic graph with a single source. Compaction is to set the X-coordinate of each block at the longest path length from the source vertex to the vertex corresponds to the block.

An extended version of constraint graph for compaction contains negative weight edges and directed cycles. (Graph contains any cycles is called "cyclic", and otherwise "acyclic".) Using edges with negative weights and cycles make it possible to represent conditions like "b should be right of a by exactly x". In this paper, rectilinear block packing will be achieved by using the concept of this extended version of constraint graph.

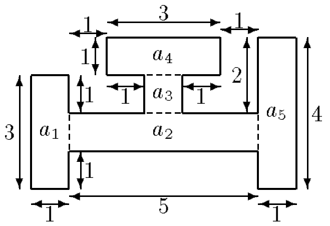


Figure 3. Rectilinear block a is partitioned into $\{a_1, a_2, a_3, a_4, a_5\}$.

If a constraint graph is cyclic, positions that satisfies all constraints imposed by the graph do not always exist. It is known that such a position exists if and only if the constraint graph contains no **positive cycles**. A "positive cycle" is a cycle whose sum of weights of edges is positive.

If a graph $G(V, E)$ is acyclic, finding the longest path length to all vertices from source vertex needs $O(|V| + |E|)$ time, using the algorithm for finding longest path in DAG (Directed Acyclic Graph) [4]. If a cyclic graph $G(V, E)$ contains no positive cycle, finding the longest path length to all vertices from source vertex needs $O((b+1)T)$ time [1] where b and T are the numbers of negative edges and all edges respectively.

There are other algorithms available to find the longest path on a cyclic graph without positive cycles. One is to apply Floyd's or Ford's shortest path algorithm [4]. How to apply the shortest path algorithm to longest path search is: Given constraint graph G, make a inverse graph G^r by inverting the sign of the weights of all edges in G, and find the shortest path length to each vertex on G^r, and inverse the sign of the path length. It is known that Floyd's algorithm can find the shortest path on graph G^r in $O\left(|V(G^r)|^3\right)$ and Ford's algorithm can find the shortest path on graph G^r in $O\left(|V(G^r)|\,|E(G^r)|\right)$ if G^r contains no negative cycles. Since negative cycles on G^r correspond exactly to positive cycles on G, applying these algorithms can find the longest path on G if and only if G contains no positive cycles. If G^r contains negative cycles, Floyd's algorithm will only fail but Ford's algorithm will confirm the presence of negative cycles before fail. So Ford's algorithm can be used to check if G would contain positive cycles.

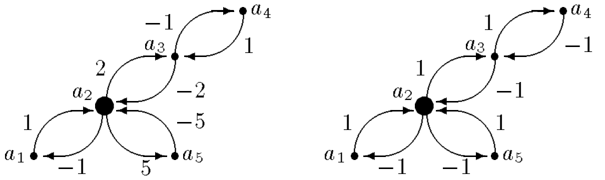
3. ALGORITHM FOR RP PROBLEM

As only rectangle blocks can be handled by seq-pair basically, rectilinear blocks are partitioned into rectangles. A rectilinear block a can be partitioned into a set of sub-blocks $\{a_1, a_2, \cdot\cdot, a_n\}$ with horizontal or vertical lines (See Fig.3), as other methods. Let $s_x(r)$ be the width of rectangle r and $s_y(r)$ be the height of r. Since we can partition a rectilinear block horizontally and vertically together, we can minimize the number of sub-blocks [2] and it leads us to save the computation time.

For simplicity, rotations and reflections of rectilinear blocks are assumed not to be permitted here. They are discussed later.

3.1. X/Y-alignment

In [13] and [17], X and Y alignment procedures are needed after rectangular packing based on the seq-pair to align sub-blocks which are the part of a common rectilinear block. But our method to align sub-blocks is different from them since the packing and the alignment are executed simultaneously. So it needs no post-process.



(a) Horizontal constraint graph G_H

(b) Vertical constraint graph G_V

Figure 4. Part of constrained graphs for the rectilinear block shown in Fig.3

In order to pack and align simultaneously, special edges called "**relative position edge pair**" are introduced. When sub-blocks a_i and a_j are adjacent parts of a common rectilinear block a, relative position edge pair, which consists of directed weighted edge (a_i, a_j) and (a_j, a_i), is introduced into both H/V constrained graphs. Directed edge (a_i, a_j) on H-constrained graph is defined from a_i to a_j, and its weight is the difference from original X-coordinate (before partitioning) of a_i to that of a_j. Directed edge (a_i, a_j) on V-constrained graph is defined from a_i to a_j, and its weight is the difference from original Y-coordinate of a_i to that of a_j. Directed edges (a_j, a_i) on H- and V-constrained graphs are defined similarly. Clearly, the sum of the weights of the edges of relative position edge pair is zero. Fig.4(a) is a part of H-constrained graphs for the rectilinear block shown in Fig.3, and Fig.4(b) is a part of V-constrained graphs.

By the compaction theorem described before, 1-D compaction using H- and V-constrained graphs which contain all relative position edge pairs for each rectilinear block may pack all rectilinear blocks without overlapping and with all alignments of rectilinear blocks if there is a packing corresponds to the constrained graphs.

3.2. Packing using constraint graphs

New horizontal and vertical constraint graphs are proposed, which make it possible to pack and align all rectilinear blocks simultaneously.

Horizontal constraint graph G_H is defined as an edge-weighted directed cyclic graph. Vertex set $V(G_H)$ consists of one source vertex and one drain vertex and vertices correspond to rectangular blocks. Edge set $E(G_H)$ can be classified into four kinds:

1. The edges from source vertex to the other, whose weights are all zero.
2. The edges to drain vertex from the other (v_i, v_d), whose weights are the width of rectangles correspond to v_i.
3. The edges (v_i, v_j) correspond to horizontal constraint imposed by the seq-pair, whose weights are the width of rectangles correspond to v_i. These edges are called "**H-constrained edges**".
4. The edges of **relative position edge pair**, which have been described before. □

Vertical constraint graph G_V is defined similarly.

Note that seq-pair constrained graphs for RP are vertex-weighted directed acyclic graphs. If there is no relative position edge pair in G_H/G_V, the weights of all the edges from one vertex are the same and the weights can move to the vertex. Then, the constrained graph can be transformed to the constrained graph defined by [5, 7] for RP.

Then, we can get an algorithm "Rectilinear block packing", using compaction on G_H and G_V below.

```
Algorithm: Rectilinear block packing
    ( input: block-sizes, seq-pair $S$ )
    ( output: packing                  )
step 1-1: Make H-constraint graph $G_H$ from $S$.
step 1-2: Determine the $X$ coordinate of each rectangle
    by the longest path length from the source vertex to the
    vertex of the block.
step 2-1: Make V-constraint graph $G_V$ from $S$.
step 2-2: Determine the $Y$ coordinate of each rectangle
    by the longest path length from the source vertex to the
    vertex of the rectangle.
```

As described in 2.3., applying the Ford's shortest path algorithm makes it possible to check the presence of positive cycles and to find the longest path simultaneously in $O(|n|^3)$ time. If Rectilinear block packing could be done, all rectilinear blocks would be aligned and packed by the compaction theorem described before.

3.3. Feasible seq-pair

If all blocks are rectangles, there always exists a packing corresponds to each arbitrary seq-pair. But if any non-rectangle blocks are contained, there may exist no packing corresponds to some seq-pair. So "feasible seq-pair" can be defined as:

Feasible sequence-pair: If there is a rectilinear block packing which keeps all H/V-constraints imposed by the seq-pair, then the seq-pair is said to be "**feasible**". And a seq-pair which is not feasible is said to be "infeasible". □

Note that in [13] and [17], "feasible seq-pair" is defined as a seq-pair which can lead to a rectilinear block packing after their post-processes, and it means the post-process is the key for the feasibility.

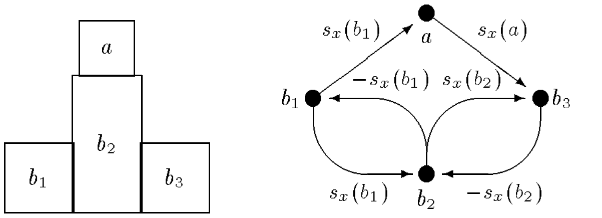
Then we can get the theorem below.

Theorem 1 *A seq-pair S is feasible if and only if both H- and V-constraint graphs of S contain no positive cycles.* □

"positive cycle" is a cycle in a directed graph and the sum of the weights of edges on the cycle is positive (bigger than zero).

Proof: If the difference from X-coordinate of rectangle a to X-coordinate of rectangle b is less than the weight of H-constrained edge (a, b), it is clear that relative positions of a and b violate the H-constraint imposed by seq-pair. And if the difference from X-coordinate of rectangle a_i to X-coordinate of rectangle a_j is less than the weight of one of relative position edge pair (a_i, a_j), rectilinear block a is not aligned on X clearly. So, if H-constrained graph contains positive cycle, rectilinear block packing could not be exist, and it means the seq-pair is infeasible. It is similar on V-constraint graph.

If both constrained graphs G_H, G_V have no positive cycle, Rectilinear block packing can find the longest path by applying Floyd's or Ford's algorithm. The packing which keeps all the constraints imposed by G_H and G_V keeps all the H- and V-constraint imposed by seq-pair S obviously, and all the rectilinear blocks would align. It means the seq-pair S is feasible. ■



(a) packed with seq-pair $(b_1ab_2b_3; b_1b_2ab_3)$

(b) Horizontal constraint graph of (a)

Figure 5. Example of difference between feasible and infeasible seq-pair: If $s_x(a) \leq s_x(b_2)$, the seq-pair is feasible. But if $s_x(a) > s_x(b_2)$, the seq-pair is infeasible.

For example, if the rectilinear block shown in Fig.3 is the only block to be packed, the feasible seq-pair is $(a_1a_4a_3a_2a_5; a_1a_2a_3a_4a_5)$ or $(a_4a_1a_3a_2a_5; a_1a_2a_3a_4a_5)$. Either G_H or G_V of another seq-pair has positive cycle.

For another example in Fig.5, rectangle a and convex rectilinear block b which is partitioned into $\{b_1, b_2, b_3\}$ are packed by seq-pair $(b_1ab_2b_3; b_1b_2ab_3)$ as shown in Fig.5(a), and horizontal constraint graph G_H of it is shown in Fig.5(b). In G_H, cycle b_1, a, b_3, b_2, b_1 whose sum of weights is $s_x(a) - s_x(b_2)$ exists. If $s_x(a) \leq s_x(b_2)$, no positive cycles exist and the seq-pair is feasible. However, if $s_x(a) > s_x(b_2)$, cycle b_1, a, b_3, b_2, b_1 is positive, and the seq-pair is infeasible. It means that feasibility of a seq-pair depends on the dimensions of rectangles. Note that if the seq-pair is $(ab_1b_2b_3; b_1b_2b_3a)$, there exist no positive cycles in G_H and the seq-pair is always feasible.

3.4. Optimality

The following theorem guarantees the optimality of the rectilinear block packing.

Theorem 2 *Rectilinear block packing with the minimum area can be represented by a seq-pair.*

Proof: From the rectilinear block packing with the minimum area P, rectangle packing P' can be got by partitioning all rectilinear blocks. Seq-pair S can be got from P' by "gridding operation" proposed in [5, 7]. H- and V-constraint graphs of S clearly contain no positive cycle, so S is feasible. ■

This theorem implies that the exhaustive search of feasible seq-pairs can find the optimum (minimum area) rectilinear block packing.

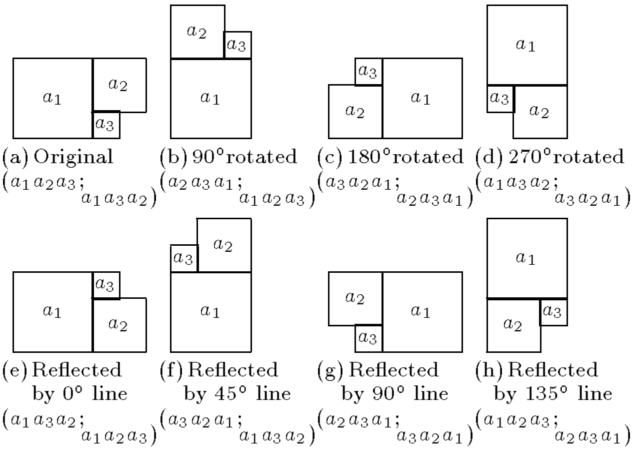
3.5. Rotation and reflection

In VLSI layout design, blocks are often permitted to reflect and rotate by 90°. Rotation of a rectangle is only an exchange of width and height, but rotation and reflection of rectilinear blocks need modification on seq-pair.

Rotation and reflection can be done with combining three basic operations of seq-pair defined follows.

Op1: Reverse the order of the former permutation (Initially, Γ_+). Then all blocks are reflected by +45° line. $\big((a_1a_2a_3; a_1a_3a_2) \rightarrow (a_3a_2a_1; a_1a_3a_2)\big)$.

Op2: Reverse the order of the latter permutation (Initially, Γ_-). Then all blocks are reflected by −45° line. $\big((a_1a_2a_3; a_1a_3a_2) \rightarrow (a_1a_2a_3; a_2a_3a_1)\big)$.



(a) Original ($a_1a_2a_3$; $a_1a_3a_2$)
(b) 90°rotated ($a_2a_3a_1$; $a_1a_2a_3$)
(c) 180°rotated ($a_3a_2a_1$; $a_2a_3a_1$)
(d) 270°rotated ($a_1a_3a_2$; $a_3a_2a_1$)
(e) Reflected by 0° line ($a_1a_3a_2$; $a_1a_2a_3$)
(f) Reflected by 45° line ($a_3a_2a_1$; $a_1a_3a_2$)
(g) Reflected by 90° line ($a_2a_3a_1$; $a_3a_2a_1$)
(h) Reflected by 135° line ($a_1a_2a_3$; $a_2a_3a_1$)

Figure 6. Variation of seq-pair with rotations and reflections.

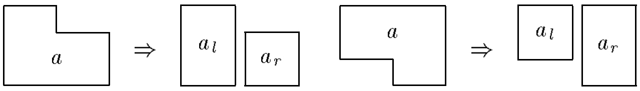


Figure 7. L-shaped block a is partitioned into a_l, a_r.

Op3: Exchange the former and the latter permutations (Initially, Γ_+ and Γ_-). Then all blocks are reflected by 0° line. $\big((a_1a_2a_3; a_1a_3a_2) \rightarrow (a_1a_3a_2; a_1a_2a_3)\big)$.

For example, the sequence of **Op1** and **Op3** makes the block rotate by 270° and the sequence of **Op3** and **Op1** makes the block rotate by 90°. The effects of the sequence of operations are displayed in Fig.6.

4. L-SHAPED BLOCK PACKING

If we limit the rectilinear block to one which can be divided into two rectangles by a vertical line, like L-shaped block (L-block), a necessary and sufficient condition for feasible seq-pair can be elegantly expressed regardless of the dimensions of rectangles. Note that L-block packing with minimum area can be clearly expressed by a feasible seq-pair.

This necessary and sufficient condition has already presented in [15]. But this condition is very important also for rectilinear block packing problem because the condition is, of course, a necessary condition for feasible seq-pair of rectilinear block packing problem. Moreover as [15] is written in Japanese, we describe the theorem for the condition here.

Suppose L-block a is partitioned by vertical line into two rectangles a_l and a_r as shown is Fig.7.

4.1. Feasible seq-pair

First, we enumerate three conditions for infeasible seq-pair.

(1) "Forbidden L-position": If a_l is not before a_r in either Γ_+ or Γ_- for any L-block a, we call this seq-pair has **Forbidden L-position**. In other words, relative position of a_l and a_r is not as

$$S = (\cdot\cdot a_l \cdot\cdot a_r \cdot\cdot ; \cdot\cdot a_l \cdot\cdot a_r \cdot\cdot).$$

(2) "L-intruder": If a rectangle x is between a_l and a_r on both Γ_+ and Γ_- for an L-block a, we call this seq-pair has **L-intruder**, or "x is a **L-intruder** of a". In other words, relative position of a_l, a_r, and x is as

$$S = (\cdot\cdot a_l \cdot\cdot x \cdot\cdot a_r \cdot\cdot ; \cdot\cdot a_l \cdot\cdot x \cdot\cdot a_r \cdot\cdot)$$

(3) "L-crossing": If both b_l and b_r are between a_l and a_r on Γ_+ and both a_l and a_r are between b_l and b_r on Γ_-, we call this seq-pair has **L-crossing**, or "a and b are **L-crossing**". In other words, relative position of a_l, a_r, b_l, and b_r is as

$$S = (\cdot\cdot, a_l, \cdot\cdot, b_l, \cdot\cdot, b_r, \cdot\cdot, a_r, \cdot\cdot ; \cdot\cdot, b_l, \cdot\cdot, a_l, \cdot\cdot, a_r, \cdot\cdot, b_r, \cdot\cdot).$$

Theorem 3 *Seq-pair S is feasible if and only if S has neither forbidden L-position nor L-intruder nor L-crossing.*

This theorem implies a feasible seq-pair of L-block packing can be represented regardless of the dimensions of rectangles. As our proof of this theorem is rather complicated, it is given separately in two subsections according to the necessary part and sufficient part.

4.2. Proof of necessary condition of Theorem 3

We will prove here that if seq-pair S has either forbidden L-position or L-intruder or L-crossing, there may be some L-blocks that cannot be aligned.

If seq-pair S has forbidden L-position at L-block a, relative position between a_l and a_r must not regular. So, the L-block a cannot be aligned obviously.

If seq-pair S has L-intruder x between a_l and a_r, rectangle x is between a_l and a_r horizontally, and a_l and a_r cannot be aligned horizontally.

As the proof for L-crossing is a little bit difficult, we discuss in the lemma below:

Lemma 1 *If seq-pair S has L-crossing between L-block a and b, relative position of either a or b cannot be aligned.*

Proof: We can decide without loss of generality that seq-pair S is

$$S = (\cdot\cdot a_l \cdot\cdot b_l \cdot\cdot b_r \cdot\cdot a_r \cdot\cdot ; \cdot\cdot b_l \cdot\cdot a_l \cdot\cdot a_r \cdot\cdot b_r \cdot\cdot).$$

If we denote the Y-coordinate of the lower line of rectangle r is $y(r)$, the Y-coordinate of the upper line of r is $y(r) + s_y(r)$. Since S implies that b_l is below a_l and a_r is below b_r, we can get:

$$y(b_l) + s_y(b_l) \leq y(a_l) \tag{1}$$

$$y(a_r) + s_y(a_r) \leq y(b_r). \tag{2}$$

Suppose L-blocks a and b could be aligned though they are L-crossing each other on S. Then, we can get:

$$y(a_l) < y(a_r) + s_y(a_r). \tag{3}$$

From equation (1) and (2) and (3), we can get

$$y(b_l) + s_y(b_l) < y(b_r). \tag{4}$$

Then, L-block b cannot be aligned, a contradiction. ■

4.3. Proof of sufficient condition of Theorem 3

We will prove here that if seq-pair S has neither forbidden L-position nor L-intruder nor L-crossing, all L-blocks can be aligned. We will prove by construction using Rectilinear block packing.

First, Rectilinear block packing is executable because **step 1-1** and **step 2-1** are obviously executable, and executability of **step 1-2** and **step 2-2** can be shown only by the proof that both H- and V-constraint graph (G_H, G_V) have no positive cycles. Proofs that G_H and G_V have no positive cycles are described in lemma 2 and lemma 3.

Path from a through $c_1, c_2, \cdot\cdot, c_m$ to b is denoted by $(a, c_1, c_2, \cdot\cdot, c_m, b)$, and cycle from a through $c_1, c_2, \cdot\cdot, c_m$ return-to a is denoted by $(a, c_1, c_2, \cdot\cdot, c_m, a)$.

Lemma 2 *Vertical constraint graph $G_V(V, E_V)$ has no positive cycles.*

Proof: In L-block packing, as L-block a is decided to be partitioned by vertical line into a_l and a_r, one of "relative position edge pair" in vertical constraint graph is from a_l to a_r and the other is from a_r to a_l. We call the edge from a_l to a_r "**right-s-edge**" and the edge from a_r to a_l "**left-s-edge**" and either of them called "**s-edge**". Also we call the edge imposed by V-constraint "**v-edge**".

Suppose that positive cycles exist in G_V. It is clear that G_V without right-s-edge and left-s-edge contains no cycles. Then, cycles on G_V contains right-s-edge or left-s-edge.

The sum of the weights of any cycles consists of right-s-edges (a_l, a_r) and left-s-edges (a_r, a_l) is exactly zero by their definitions. Hence the positive cycle contains some v-edges if exists.

Now, we call one of the cycles with the minimum length that includes v-edge $\mathcal{L}$. (Note that we do not mind whether the sum of the weights of $\mathcal{L}$ is positive or not.) In the following, we will show that $\mathcal{L}$ must not exist.

It is obvious from its definition that start vertex and end vertex of two consecutive s-edges are the same, and that v-edges are transitive. So, $\mathcal{L}$ is alternate s-edge and v-edge.

If we compare start vertex to end vertex of three kinds of edges Γ_+ decreases and Γ_- increases in v-edges, with respect to the order in Γ_+ and Γ_-. and both Γ_+ and Γ_- increase in right-s-edges, and both Γ_+ and Γ_- decrease in left-s-edges. Hence there exist no cycles with only v-edges and left-s-edges, and there exist no cycles with only v-edges and right-s-edges. $\mathcal{L}$ must contain at least one v-edge and at least one left-s-edge and at least one right-s-edge.

Then there may be a v-edge on $\mathcal{L}$ whose predecessor is a right-s-edge, and whose successor is a left-s-edge. We assume this path is (a_l, a_r, b_r, b_l). (Edge (a_l, a_r) is a right-s-edge, (a_r, b_r) is a v-edge, and (b_r, b_l) is a left-s-edge.) From the conditions for three kinds of edges, we can get

$$\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(a_r) > \Gamma_+^{-1}(b_r) > \Gamma_+^{-1}(b_l) \qquad (5)$$

$$\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(a_r) < \Gamma_-^{-1}(b_r) > \Gamma_-^{-1}(b_l) \qquad (6)$$

Note that none of v-edge (a_l, b_r), (a_l, b_l), and (a_r, b_l) exist because $\mathcal{L}$ is a cycle with minimum length. Since edge (a_l, b_r) never exists, it is straightforward that $\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_r)$. See Fig.8 for intuitive understanding.

If $\Gamma_-^{-1}(b_l) > \Gamma_-^{-1}(a_r)$ (b_l is in domain A or B on Fig.8), v-edge (a_r, b_l) exists, a contradiction. Hence $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(a_r)$.

Here, we can classify the relative positions between a_l and b_l into four kinds.

If $\underline{\Gamma_+^{-1}(b_l) < \Gamma_+^{-1}(a_l) \text{ and } \Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b_l)}$, ($b_l$ is in domain C on Fig.8), v-edge (a_l, b_l) exists, a contradiction.

If $\underline{\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_l) \text{ and } \Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b_l)}$, ($b_l$ is in domain D on Fig.8), b_l is an L-intruder of a_r, a_l, a contradiction.

If $\underline{\Gamma_+^{-1}(b_l) < \Gamma_+^{-1}(a_l) \text{ and } \Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(a_l)}$, ($b_l$ is in domain E on Fig.8), a_l is an L-intruder of b_l, b_r, a contradiction.

If $\underline{\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_l) \text{ and } \Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(a_l)}$, ($b_l$ is in domain F on Fig.8), a and b are L-crossing, a contradiction. ■

Next, we discuss about the existence of positive cycles on H-constraint graph G_H. There is a cycle with length two which corresponds to each L-block on G_H by the same reason on G_V, and the sum of the weights of edges on the cycle is all zero. There exist some cycles with more than two edges is different from the case of G_V. For example, in H-constraint graph of $S = (a_l, b_l, a_r, b_r; b_l, a_l, b_r, a_r)$, there exists a cycle $(a_r, a_l, b_r, b_l, a_r)$, but the sum of the weights of the edges of this cycle is also zero. The following lemma holds.

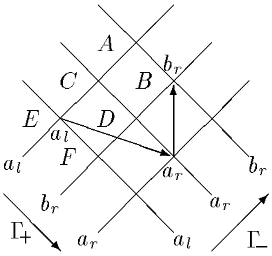


Figure 8. Existence of cycle $\mathcal{L}$ on V-constraint graph G_V

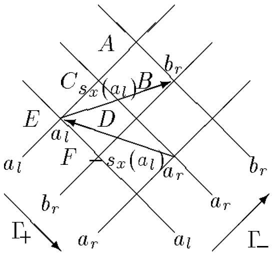


Figure 9. Existence of positive cycle C on H-constraint graph G_H

Lemma 3 *Horizontal constraint graph $G_H(V, E_H)$ has no positive cycles.*

Proof: We call the edge imposed by H-constraint "**h-edge**". In L-block packing, as L-block a is decided to be partitioned by vertical line into a_l and a_r, one of "relative position edge pair" in horizontal constraint graph is from a_l to a_r and the other is from a_r to a_l. We call the edge from a_r to a_l "**reverse-edge**" The edge from a_l to a_r is redundant because there exists an h-edge (a_l, a_r) with the same weight.

Suppose that positive cycles exist in G_H. We call one of the positive cycles with the minimum length C. In the following, we will show that C must not exist.

It is clear that G_H without reverse-edges contains no cycles. Then, C contains any reverse-edges.

Pay attention to a reverse-edge (a_r, a_l) on C. The next vertex after a_l on C is called b. Check the relative positions of b among a_l and a_r on Γ_+ and Γ_-. First, (a_l, b) is an h-edge because the predecessor on C is a reverse-edge, and reverse-edges cannot adjoin by definition. Hence $\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b)$. Here, we can classify the relative positions between b and a_r on Γ_+ and Γ_- into five kinds.

If $\underline{b \text{ is equal to } a_r}$, cycle (a_r, a_l, a_r) exists. C is exactly $\underline{\text{equal to } (a_r, a_l, a_r)}$ because C never pass a vertex twice. The sum of the weights of the edges on this cycle is $-s_x(a_l) + s_x(a_l) = 0$, not positive, a contradiction.

If $\underline{b \text{ is left to } a_r}$ ($\Gamma_+^{-1}(a_r) > \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a_r) > \Gamma_-^{-1}(b)$), $\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b) < \Gamma_+^{-1}(a_r)$ and $\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b) < \Gamma_-^{-1}(a_r)$. b is an L-intruder of a_l, a_r, a contradiction.

If $\underline{b \text{ is right of } a_r}$ ($\Gamma_+^{-1}(a_r) < \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a_r) < \Gamma_-^{-1}(b)$), there exists h-edge (a_r, b). Assume $C = (a_r, a_l, b, \cdot\cdot, a_r)$, and replace (a_r, a_l, b) on C with (a_r, b). Compare the sum of the weights on cycle $C' = (a_r, b, \cdot\cdot, a_r)$ and that on C. The sum of the weights of edges (a_r, a_l) and (a_l, b) is $-s_x(a_l) + s_x(a_l) = 0$, and the weight of h-edge (a_r, b) is positive. And the sum of the weights of the edges on C' is bigger than that on C. Then, C' is positive weight and fewer number of vertices than C, a contradiction.

If $\underline{b \text{ is above } a_r}$ ($\Gamma_+^{-1}(a_r) > \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a_r) < \Gamma_-^{-1}(b)$) can occur. Path (a_r, a_l, b) is called "**up-path**". The sum of the weights of the edges on up-path (a_r, a_l, b) is trivially zero by definition.

If b is below a_r $(\Gamma_+^{-1}(a_r) < \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a_r) > \Gamma_-^{-1}(b))$ can occur. Path (a_r, a_l, b) is called "**down-path**. The sum of the weights of the edges on down-path (a_r, a_l, b) is trivially zero by definition.

As there comes an h-edge next to a reverse-edge on C, C can be divided into a set of h-edges S_H and a set of up-paths and down-paths S_{ud}. Since the sum of the weights of an element of S_{ud} is all zero, and the weight of an element of S_H is all positive, and the sum of the weights of edges on C is positive, $|S_H| > 0$. Otherwise, the sum of the weights of edges on C would be zero.

The predecessor of one of the h-edges e_h in S_H is an h-edge e_p because the latter edge of both up-path and down-path is an h-edge. As h-edges are transitive, a cycle C' can be made by combining e_p and e_h. If we assume $|S_H| \geq 2$, C' would be a positive cycle and made of fewer vertices, a contradiction. (Note that if $|S_H| = 1$, the sum of the weights of the edges on C' would be zero, no contradiction.) Hence $|S_H| = 1$.

If we compare start vertex to end vertex of h-edge and up-path and down-path, with respect to the order in Γ_+ and Γ_-, both Γ_+ and Γ_- increase in h-edges, and Γ_+ decreases and Γ_- increases in up-path, and Γ_+ increases and Γ_- decreases in down-path. Hence any cycle contains both up-paths and down-paths.

From the reasons before, C consists of exactly one h-edge, and one or more than one up-paths, and one or more than one down-paths. Then, there exists a point either a down-path follows a up-path, or a up-path follows a down-path.

Without loss of generality, we assume up-path (a_r, a_l, b_r) is followed by down-path (b_r, b_l, c) on C. Then,

$$\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_r) < \Gamma_+^{-1}(a_r) \qquad \text{and} \tag{7}$$

$$\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(a_r) < \Gamma_-^{-1}(b_r). \tag{8}$$

Now, we check the position of b_l on Γ_+ and Γ_-. $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(b_r)$ and $\Gamma_+^{-1}(b_l) < \Gamma_+^{-1}(b_r)$ because (b_r, b_l) is a reverse-edge. If $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(a_l)$ and $\Gamma_+^{-1}(b_l) < \Gamma_+^{-1}(a_l)$ (b_l is in domain E on Fig.9), a_l is an L-intruder of b_r, b_l, a contradiction. Else if $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(a_l)$ and $\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_l) < \Gamma_+^{-1}(b_r)$! Jb_l is in domain F on Fig.9), a and b are L-crossing, a contradiction. Hence,

$$\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b_l) \tag{9}$$

! Jb_l is in domain A, B, C, or D on Fig.9).

Next, we check the position of c on Γ_+ and Γ_-. As (b_r, b_l, c) is a down-path, $\Gamma_+^{-1}(b_r) < \Gamma_+^{-1}(c)$. Also from equation (7),

$$\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_r) < \Gamma_+^{-1}(c) \tag{10}$$

As (b_l, c) is an h-edge, $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(c)$. Also from equation (9),

$$\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(c) \tag{11}$$

Then, from equation (10) and (11), c is right of a_l and h-edge (a_l, c) exists. Also, C includes (a_r, a_l, b_r, b_l, c). By skipping b_r and b_l using h-edge (a_l, c), we can get a cycle C'. Since the sum of the weights of the edges on down-path (b_r, b_l, c) is zero and the weight of edge (a_l, b_r) and (a_l, c) is the same, the sum of weights of edges on C' is the same to that on C. Hence, C' is a positive cycle with fewer vertices than C by two, a contradiction. ■

Table 1. All seq-pairs of rectangle a and L-block b which do not have forbidden L-position.

sequence-pair	feasible ?	reachable group
$(a\,b_l\,b_r\ ;\ a\,b_l\,b_r)$	Yes	group 1
$(a\,b_l\,b_r\ ;\ b_l\,a\,b_r)$	Yes	group 1
$(a\,b_l\,b_r\ ;\ b_l\,b_r\,a)$	Yes	group 2
$(b_l\,a\,b_r\ ;\ a\,b_l\,b_r)$	Yes	group 1
$(b_l\,a\,b_r\ ;\ b_l\,a\,b_r)$	No (L-intruder)	—
$(b_l\,a\,b_r\ ;\ b_l\,b_r\,a)$	Yes	group 2
$(b_l\,b_r\,a\ ;\ a\,b_l\,b_r)$	Yes	group 1
$(b_l\,b_r\,a\ ;\ b_l\,a\,b_r)$	Yes	group 2
$(b_l\,b_r\,a\ ;\ b_l\,b_r\,a)$	Yes	group 2

$$\begin{tikzcd}
(a\,b_l\,b_r\ ;\ a\,b_l\,b_r) \arrow[d, Leftrightarrow, "\substack{\text{half-exchange}\\ a \text{ and } b_l}"] & \\
(b_l\,a\,b_r\ ;\ a\,b_l\,b_r) \arrow[r, Leftrightarrow, "\text{full-exchange}", "a \text{ and } b_l"'] \arrow[d, Leftrightarrow, "\substack{\text{half-exchange}\\ a \text{ and } b_r}"] & (a\,b_l\,b_r\ ;\ b_l\,a\,b_r) \\
(b_l\,b_r\,a\ ;\ a\,b_l\,b_r) &
\end{tikzcd}$$

Figure 10. Reachable seq-pair from $(a\,b_l\,b_r\ ;\ a\,b_l\,b_r)$ by half-exchange and full-exchange

Now, we can conclude that Rectilinear block packing is proved to be executable. Time complexity of Rectilinear block packing is dominated by the longest path search at **step 1-2** and **step 2-2**, and is $O(n^3)$ where n is the number of rectangles. (See 2.3. in detail.)

5. EXPERIMENTAL RESULTS

The proposed algorithm is implemented using C language, and tested on SUN Ultra-SPARC workstation. The Ford's shortest path algorithm is used to check the positive cycles and to find the longest path on horizontal and vertical constraint graphs. The solution space is searched by simulated annealing method. Two kinds of adjacent seq-pair is used:

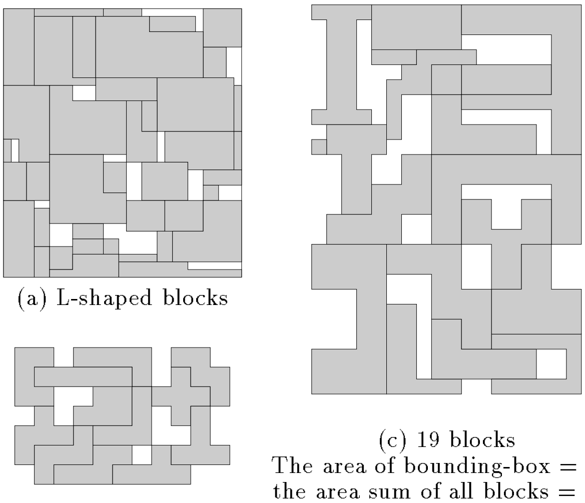
Γ_+-exchange: exchange of two rectangles on Γ_+, and

Γ_--exchange: exchange of two rectangles on Γ_-.

If a randomly made adjacent seq-pair is infeasible, the seq-pair is canceled and another adjacent seq-pair is made until the seq-pair is feasible. (As this method to get an adjacent seq-pair is time consuming, a solution space that consists of feasible seq-pairs only or that uses adaptation technique [10] is preferable.)

In [5, 7], "full-exchange", which exchanges two rectangles on both Γ_+ and Γ_-, and "half-exchange", which is the same as "Γ_+-exchange" are defined to use for making adjacent seq-pairs. But these adjacent seq-pair definitions are proved easily that a certain seq-pair is not reachable from another certain seq-pair. In Table 1, all seq-pairs of rectangle a and L-block b which do not have forbidden L-position are listed. All the seq-pairs, except for an infeasible (L-intruder) one, are classified into group 1 and group 2, and a seq-pair of group 1 is not reachable from a seq-pair of group 2. Adjacent seq-pairs in group 1 is displayed in Fig.10. (Note that solution space with Γ_+ and Γ_--exchange is not proved to be reachable from a seq-pair to any other seq-pair.)

Fig.11(a) is a packing example of the same data as Fig.10 in [6], that consists of 10 L-blocks and 30 rectangles. (It is



(a) L-shaped blocks

(b) Pentomino

(c) 19 blocks
The area of bounding-box = 468
the area sum of all blocks = 353
CPU time = 13446[s]

Figure 11. Packing examples of rectilinear blocks

reflected by X-axis.) This packing without block rotations and reflections takes 5 minutes.

Fig.11(b) is a packing example of 12 rectilinear blocks, which includes a concave rectilinear block. These blocks are known as "pentomino". Fig.11(c) is a packing example of 19 rectilinear blocks, which is randomly handmade.

6. CONCLUSIONS

We have proposed a new method to pack a set of rectilinear blocks allowing concave blocks, which has not been considered, based on the seq-pair. It has been proved that any packing can be represented by it, hence the packing with minimum area. A necessary and sufficient condition for feasible seq-pair of rectilinear block packing have been proposed and proved, And a necessary and sufficient condition for seq-pair of L-shaped blocks also have been proposed and proved regardless of the dimensions of rectangles. Experimental results show effectiveness of the proposed method.

The proposed algorithm Rectilinear block packing uses computational complexity of $O(n^3)$. However, it could be improved to work in $O(n^2 \log n)$. The proposed method is useful not only for the packing of rectilinear blocks but also for a group of blocks with fixed relative positions, which are seen in PCB design problem often. Also a group of blocks whose relative positions are limited in some extents can be represented.

ACKNOWLEDGMENT

The authors would like to thank Prof Y. Kajitani and Prof A. Takahashi, Tokyo Institute of Technology, for helpful comments. This work was supported in part by Research Body CAD21 and a Grant in Aid for Scientific Research of the Ministry of Education, Science, and Culture of Japan

REFERENCES

[1] Y-Z. Liao and C. K. Wong: "An Algorithm to Compact a VLSI Symbolic Layout with Mixed Constraints", IEEE Trans. on CAD, Vol.2, No.2, pp.62–69 (1983).

[2] T. Ohtsuki, M. Sato, M. Tachibana, and S. Torii: "Minimum Partitioning of Rectilinear Regions", Trans. IPSJ, Vol.24, No.5, pp.647–653 (1983).

[3] T. Yoshimura: "A Graph Theoretical Compaction Algorithm", in Proc. ISCAS'85, pp.1455–1458 (1985).

[4] J. A. McHugh: "Algorithmic Graph Theory", Prentice Hall (1990).

[5] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani: "Rectangle-Packing-Based Module Placement", Proc. ICCAD'95, pp.472–479 (1995).

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

[7] 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 (1996).

[8] M. Kang, and W. W.-M. Dai: "General Floorplanning with L-shaped, T-shaped and Soft Blocks Based on Bounded Slicing Grid Structure", in Proc. ASP-DAC'97, pp.265–270 (1997).

[9] J. Dufour, R. McBride, P. Zhang, and C.-K. Cheng: "A building Block Placement Tool", in Proc. ASP-DAC'97, pp.271–276 (1997).

[10] H. Murata, K. Fujiyoshi, M. Kaneko: "VLSI/PCB Placement With Obstacles Based on Sequence-Pair", 1997 Intl. Symp. on Physical Design, pp.26-31, Apr. (1997).

[11] H. Murata, K. Fujiyoshi, M. Kaneko: "VLSI/PCB Placement With Obstacles Based on Sequence-Pair", IEEE Trans. on CAD, Vol.17, No.1, pp.60–68 (1998).

[12] S. Nakatake, M. Furuya, and Y. Kajitani: "Module Placement on BSG-Structure with Preplaced Modules and Rectilinear Modules", in Proc. ASP-DAC'98, pp.571–576 (1998).

[13] J. Xu, P.-N. Guo, and C.-K. Cheng: "Rectilinear Block Placement Using Sequence-Pair", in International Symposium of Physical Design, pp.173–178 (1998).

[14] M. Z.-W. Kang and W. W.-M. Dai: "Topology Constrained Rectilinear Block Packing", in International Symposium of Physical Design, pp.179–186 (1998).

[15] K. Fujiyoshi and H. Murata: "A Method to Pack L-shaped and Rectangular Modules Using Sequence-Pair", The 11-th Workshop on Circuits and Systems in Karuizawa, pp.113–118, April (1998).

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

[17] M. Z. Kang and W. W.-M. Dai: "Arbitrary Rectilinear Block Packing Based on Sequence Pair", in Proc. ICCAD'98, pp.259–266 (1998).

[18] K. Sakanushi, S. Nakatake, and Y. Kajitani: "The Multi-BSG: Stochastic Approach to an Optimum Packing of Convex-Rectilinear Blocks", in Proc. ICCAD'98, pp.267–274 (1998).