

# Arbitrary Convex and Concave Rectilinear Block Packing Using Sequence-Pair

Kunihiro Fujiyoshi, *Member, IEEE* and Hiroshi Murata, *Associate Member, IEEE*

**Abstract**—The sequence-pair was proposed in 1995 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 unified way to represent the packing of a set of rectilinear blocks, including arbitrary concave rectilinear blocks. Our idea is in the 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. If we limit the rectilinear blocks to L-shaped ones, a necessary and sufficient condition can be represented only in terms of the topology of the sequence-pair, without dimensions of blocks. A packing algorithm is designed as an SA search of the generated sequence-pairs. Experimental results show effectiveness of the proposed method.

**Index Terms**—Compaction, layout, packing, placement, sequence-pair.

## I. INTRODUCTION

RECENT advance of submicron technology makes it possible to realize a big system on a single chip. To design such a huge VLSI layout, 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 (convex blocks)** and **concave rectilinear blocks (concave 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], [8] and BSG [7], [17] 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 a 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 [10], [14], [16], [18] and by BSG [7], [9], [13], [15], [17], [19]. The only method that can pack concave rectilinear blocks is [13]. However, [7], [9], [10], [13], [14], and [17] 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, if the blocks are only convex, it is known that the condition for the set of seq-pair which can represent any packing but does not include a certain set of feasible seq-pair (see Fig. 5) may not include dimensions of blocks [18]. Moreover, if the blocks are L-shaped and rectangular, the condition for feasible seq-pair is simplified so as not to use dimensions of blocks [16], [20].

The rest of the paper is organized as follows. In Section II, we review some important research results. Section III is devoted for rectilinear block packing problem. In Section IV, a necessary and sufficient condition for a feasible seq-pair of L-shaped blocks is discussed. We consider on the solution space of rectilinear block packings in Section V. In Section VI, some experiments are demonstrated. Section VII is conclusions.

## II. PRELIMINARY

### A. Sequence-Pair

A seq-pair [5], [8] 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  $(n!)^2$ .

Manuscript received June 1, 1999; revised October 8, 1999. This work was supported in part by Research Body CAD21 and in part by the Ministry of Education, Science, and Culture of Japan under a Grant in Aid for Scientific Research. This paper was recommended by Associate Editor M. Wong.

K. Fujiyoshi is with the Department of Electrical and Electronic Engineering, Tokyo University of Agriculture and Technology, Koganei-shi, Tokyo 184-8588, Japan.

H. Murata is with MicroArk Company, Ltd., Tokyo 104-0061, Japan.

Publisher Item Identifier S 0278-0070(00)01797-8.

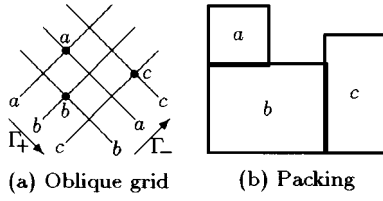


Fig. 1. Example of oblique-grid and packing of seq-pair  $(abc; bac)$ .

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

$$\begin{aligned}
 (\dots a \dots b \dots; \dots a \dots b \dots) &\rightarrow \begin{cases} \text{the right side of } a \text{ is} \\ \text{left of the left side of } b \\ (a \text{ is left of } b) \end{cases} \\
 (\dots b \dots a \dots; \dots a \dots b \dots) &\rightarrow \begin{cases} \text{the upper side of } a \text{ is} \\ \text{below the lower side of } b \\ (a \text{ is below } b). \end{cases}
 \end{aligned}$$

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

The H/V constraints of a seq-pair can be intuitively grasped using the **oblique-grid** notation. 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. For example, Fig. 1(a) shows the oblique grid of seq-pair  $(abc; bac)$ . It shows the H/V constraints: block  $c$  is in the right quarter view range (between  $-45^\circ$  and  $+45^\circ$ ) of block  $a$  on the oblique grid, then  $c$  should be placed to the right of  $a$ .

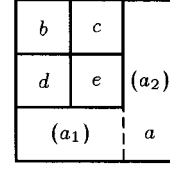
It has been proved in [5] and [8] that the set of H/V constraints imposed by every seq-pair is feasible, and an area minimum packing under the constraint can be obtained in polynomial time, and further, there is a seq-pair which leads an (globally) area minimum packing. Then, the seq-pair is easily utilized as a coding scheme of a stochastic algorithm.

To construct an area minimum placement for a seq-pair, one-dimensional (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.

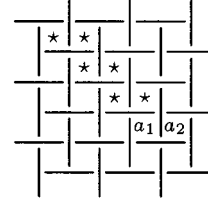
The realization can be obtained in  $O(n^2)$  time by using the **H/V constraint graph** which is constructed faithfully to the H/V constraints. More in detail, **Step 1** constructs a vertex weighted directed acyclic graph whose vertex set corresponds to the blocks and whose edge set corresponds to 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.

### B. Previous Researches for RP Problem

In [7] and [17], a method to pack L-shaped blocks based on BSG is proposed: Partition each L-shaped block into two



(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  $*$

Fig. 2. Example where BSG cannot represent the optimum packing.

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 class of L-shaped block packings. 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  $b, c, d$ , and  $e$  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  $*$  in Fig. 2(b)] But in these limited BSG-rooms, relative positions of  $b, c, d$ , and  $e$  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 more complicated method to pack L-shaped and T-shaped blocks using BSG is proposed in [9], but by the same reason with [7], [17], a certain class of packings including an example shown in Fig. 2(a) cannot be represented. In [10], a seq-pair-based method using a complicated relationship to pack rectilinear blocks is proposed, but overlaps may occur because their way of listing constraints is not enough to avoid overlaps, see also the comment in [15].

In [13], 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 [14], 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 mound-shaped rectilinear blocks because other shaped blocks may need “expansion,” which is a deformation of a block, as the authors point out.

In [15], a topology-based convex rectilinear block packing using seq-pair and BSG structure is proposed. However, the topology of a certain class of initial packings, [for example, Fig. 2(a)] cannot be represented by the same reason with [7],

[17], and so the blocks may not be packed. (“Topological compaction” can be done easily, by using the proposed method of this paper.)

In [16] and [20], a method based on the seq-pair that can represent any L-shaped block packings is proposed. Feasible seq-pair is defined as a seq-pair which corresponds to any packing, and a necessary and sufficient condition for the feasible seq-pair is shown, without the dimensions of blocks.

In [18], a method based on the seq-pair that can represent any convex rectilinear block packings is proposed. Procedures called X-, Y-alignment are proposed to align rectilinear blocks. A necessary and sufficient condition for a seq-pair with which all blocks can be aligned by the procedures is given, without the dimensions of blocks. Note that only the convex rectilinear blocks are in consideration contrary to the title.

In [19], a BSG-based convex block packing method which treats each 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.

### C. 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.

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 directed acyclic graph (DAG) [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 as follows. Given constraint graph  $G$ , make an inverse graph  $G^r$  by inverting the sign of the weights of all edges in  $G$ , and find the shortest path length

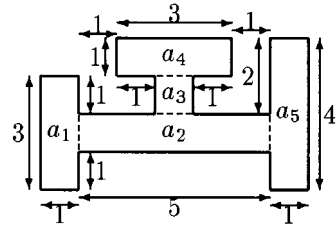


Fig. 3. Rectilinear block  $a$  is partitioned into  $\{a_1, a_2, a_3, a_4, a_5\}$ .

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  $O(|V(G^r)|^3)$  time and Ford's algorithm can find the shortest path on graph  $G^r$  in  $O(|V(G^r)||E(G^r)|)$  time 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.

## III. 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 subblocks  $\{a_1, a_2, \dots, a_n\}$  with horizontal or vertical lines (See Fig. 3), as other methods. Since we can partition a rectilinear block horizontally and vertically together, we can minimize the number of subblocks by [2] and it leads us to save the computation time. Let  $s_x(r)$  be the width of rectangle  $r$  and  $s_y(r)$  be the height of  $r$ .

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

### A. X/Y-Alignment

In [14] and [18], X and Y alignment procedures are needed after rectangular packing based on the seq-pair to align subblocks, which are the part of a common rectilinear block. Our method to align subblocks is different from them since the packing and the alignment are executed simultaneously. So it needs no post-process.

In order to pack and align simultaneously, special edges called “**relative position edge pair**” are introduced. When subblocks  $a_i$  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 constraint graphs. The weight of a directed edge  $(a_i, a_j)$  on H-constraint graph is defined as the difference from original X-coordinate (before partitioning) of  $a_i$  to that of  $a_j$ . The weight of a directed edge  $(a_i, a_j)$  on V-constraint graph is defined as the difference from original Y-coordinate of  $a_i$  to that of  $a_j$ . The weight of directed edges  $(a_j, a_i)$  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-constraint graphs for the rectilinear block shown in Fig. 3, and Fig. 4(b) is a part of V-constraint graphs.

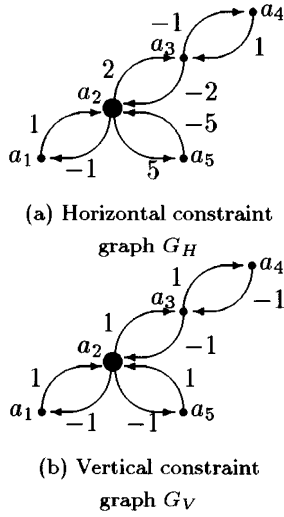


Fig. 4. Part of constraint graphs for the rectilinear block shown in Fig. 3.

By the compaction theorem described before, 1-D compaction using H- and V-constraint 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 constraint graphs.

### B. 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 types.

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

**Vertical constraint graph  $G_V$**  is defined similarly.

Note that the constraint graphs of a seq-pair 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 constraint graph can be transformed to the constraint graph defined by [5] and [8] for RP.

Then, we can get an algorithm "Rectilinear block packing," using compaction on  $G_H$  and  $G_V$  as follows:

### Algorithm: Rectilinear block packing

(input: block sizes, seq-pair  $S$ ,  
output: packing)

- Step 1) Make H-constraint graph  $G_H$  from  $S$ .
- Step 2) Determine the  $X$  coordinate of each rectangle by the longest path length from the source vertex to the vertex of the block.
- Step 3) Make V-constraint graph  $G_V$  from  $S$ .
- Step 4) 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 Section II-C, 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.

### C. Feasible Seq-Pair

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

**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 [14] and [18], "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 following theorem.

**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 both constraint graphs  $G_H$  and  $G_V$  have no positive cycle, procedure "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.

Suppose there is a rectilinear block packing  $P$  which keeps all H-constraints imposed by  $S$  although H-constraint graph of  $S$  contains a positive cycle. Obviously, there is at least one edge  $(a, b)$  in the cycle where the difference from the X-coordinate of rectangle  $a$  to the that of rectangle  $b$  is less than the weight of  $(a, b)$  in  $P$ . If  $(a, b)$  is a constraint edge imposed by  $S$ , it is clear that relative positions of  $a$  and  $b$  violate the H-constraint imposed by seq-pair. Otherwise,  $(a, b)$  is from relative position edge pair. Then, rectilinear block which contains  $a$  and  $b$  is not aligned on X clearly, a contradiction. So, if H-constraint graph contains positive cycle, there exist no rectilinear block packing, and it means the seq-pair is infeasible. It is similar on V-constraint graph.  $\square$

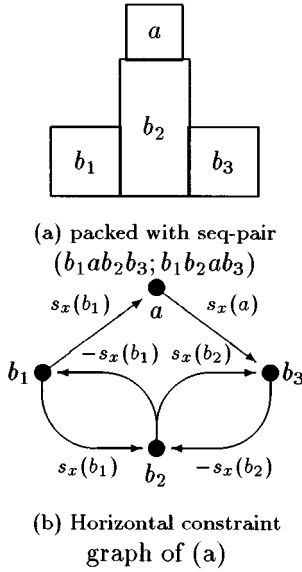


Fig. 5. Example of difference between feasible and infeasible seq-pair: If  $s_x(a) \leq s_x(b_2)$ , the seq-pair is feasible. But if  $s_x(a) > s_x(b_2)$ , the seq-pair is infeasible.

For example, if the rectilinear block shown in Fig. 3 is the only block to be packed, the feasible seq-pair is  $(a_1 a_4 a_3 a_2 a_5; a_1 a_2 a_3 a_4 a_5)$  or  $(a_4 a_1 a_3 a_2 a_5; a_1 a_2 a_3 a_4 a_5)$ . Either  $G_H$  or  $G_V$  of another seq-pair has positive cycles.

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  $(b_1 a b_2 b_3; b_1 b_2 a b_3)$  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  $s_x(a) > s_x(b_2)$ , the 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  $(a b_1 b_2 b_3; b_1 b_2 b_3 a)$ , there exist no positive cycles in  $G_H$  and the seq-pair is always feasible.

#### D. Optimality

The following theorem guarantees that any rectilinear block packing can be represented by a feasible seq-pair, including the area minimum packings.

**Theorem 2:** Any rectilinear block packing can be represented by a seq-pair.

*Proof:* From a rectilinear block packing  $P$ , rectangle packing  $P'$  can be obtained by partitioning all rectilinear blocks. Seq-pair  $S$  can be got from  $P'$  by “gridding operation” proposed in [5], [8]. H- and V-constraint graphs of  $S$  clearly contain no positive cycle, so  $S$  is feasible.  $\square$

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

### IV. L-SHAPED BLOCK PACKING

If blocks are only rectangular and L-shaped (L-block), a necessary and sufficient condition for feasible seq-pair can be el-

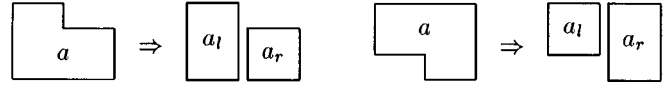


Fig. 6. L-shaped block  $a$  is partitioned into  $a_l, a_r$ .

egantly expressed regardless of the dimensions of rectangles [16], [20]. Note that any L-block packing can be expressed by a feasible seq-pair because of Theorem 2.

The reason why the condition can be expressed regardless of the dimensions is that an L-block can be partitioned into two rectangles by a line. If some blocks are partitioned into three or more rectangles, a necessary and sufficient condition for feasible seq-pair cannot be expressed without the dimensions of blocks. The reason is: It is obvious for concave blocks. For convex blocks, one example is shown in Fig. 5. (If the blocks are only convex, it is known that the condition for the set of seq-pair which can represent any packing but does not include a certain set of feasible seq-pair may not include dimensions of blocks [18].)

This necessary and sufficient condition has been already presented in [16], [20], but we will study in the following for unified theory. Note that the condition is, of course, a necessary condition for feasible seq-pair of rectilinear block packing problem, and checking the condition for L-block packing needs  $O(n^2)$  time, faster than that for convex blocks, which needs  $O(n^3)$  time.

Suppose L-block  $a$  is partitioned by vertical line into two rectangles  $a_l$  and  $a_r$  as shown in Fig. 6.

#### A. 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  $\Gamma_+$  or  $\Gamma_-$  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

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

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

$$S = (\dots a_l \dots x \dots a_r \dots; \dots a_l \dots x \dots a_r \dots).$$

- 3) **“L-crossing”**: If both  $b_l$  and  $b_r$  are between  $a_l$  and  $a_r$  on  $\Gamma_+$  and both  $a_l$  and  $a_r$  are between  $b_l$  and  $b_r$  on  $\Gamma_-$ , 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

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

**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 subsections IV-B and IV-C according to the necessary part and sufficient part.

### B. 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  $x$  exists 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 following lemma:

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

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

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

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 obtain the following:

$$y(b_l) + s_y(b_l) \leq y(a_l) \quad (1)$$

$$y(a_r) + s_y(a_r) \leq y(b_r). \quad (2)$$

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

$$y(a_l) < y(a_r) + s_y(a_r). \quad (3)$$

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

$$y(b_l) + s_y(b_l) < y(b_r). \quad (4)$$

Then, L-block  $b$  cannot be aligned, a contradiction.  $\square$

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

**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  $\Gamma_+$  decreases and  $\Gamma_-$  increases in v-edges, with respect to the order in  $\Gamma_+$  and  $\Gamma_-$ . and both  $\Gamma_+$  and  $\Gamma_-$  increase in right-s-edges, and both  $\Gamma_+$  and  $\Gamma_-$  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

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

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

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. 7 for intuitive understanding.

If  $\Gamma_-^{-1}(b_l) > \Gamma_-^{-1}(a_r)$  ( $b_l$  is in domain  $A$  or  $B$  on Fig. 7), 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 types.

If  $\Gamma_+^{-1}(b_l) < \Gamma_+^{-1}(a_l)$  and  $\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b_l)$ , ( $b_l$  is in domain  $C$  on Fig. 7), v-edge  $(a_l, b_l)$  exists, a contradiction.

If  $\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_l)$  and  $\Gamma_-^{-1}(a_l) < \Gamma_-^{-1}(b_l)$ , ( $b_l$  is in domain  $D$  on Fig. 7),  $b_l$  is an L-intruder of  $a_r, a_l$ , a contradiction.

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. 7),  $a_l$  is an L-intruder of  $b_l, b_r$ , a contradiction.

If  $\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_l)$  and  $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(a_l)$ , ( $b_l$  is in domain  $F$  on Fig. 7),  $a$  and  $b$  are L-crossing, a contradiction.  $\square$

Next, we discuss about the existence of positive cycles on H-constraint graph  $G_H$ . There is a cycle with length two which

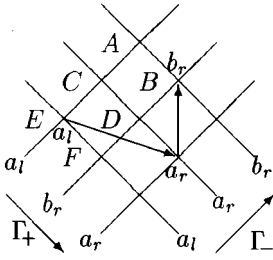


Fig. 7. Existence of cycle  $\mathcal{L}$  on V-constraint graph  $G_V$ .

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.

**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  $\Gamma_+$  and  $\Gamma_-$ . 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  $\Gamma_+$  and  $\Gamma_-$  into five kinds.

If  $b$  is equal to  $a_r$ , cycle  $(a_r, a_l, a_r)$  exists.  $C$  is exactly 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  $b$  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  $b$  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  $C' = (a_r, a_l, b, \dots, a_r)$ , and replace  $(a_r, a_l, b)$  on  $C$  with  $(a_r, b)$ . Compare the sum of the weights on cycle  $C' = (a_r, b, \dots, a_r)$  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  $b$  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  $\Gamma_+$  and  $\Gamma_-$ , both  $\Gamma_+$  and  $\Gamma_-$  increase in h-edges, and  $\Gamma_+$  decreases and  $\Gamma_-$  increases in up-path, and  $\Gamma_+$  increases and  $\Gamma_-$  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

$$\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_r) < \Gamma_+^{-1}(a_r) \quad (7)$$

and

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

Now, we check the position of  $b_l$  on  $\Gamma_+$  and  $\Gamma_-$ .  $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(b_r)$  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. 8),  $a_l$  is an L-intruder of  $b_r, b_l$ , a contradiction. Else if  $\Gamma_-^{-1}(b_l) < \Gamma_-^{-1}(a_l)$  and  $\Gamma_+^{-1}(a_l) < \Gamma_+^{-1}(b_l) < \Gamma_+^{-1}(b_r)$  ( $b_l$  is in domain  $F$  on Fig. 8),  $a$  and  $b$  are L-crossing, a contradiction. Hence

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

( $b_l$  is in domain  $A, B, C$ , or  $D$  in Fig. 8).

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

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

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

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

Then, from (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

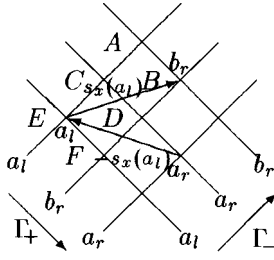
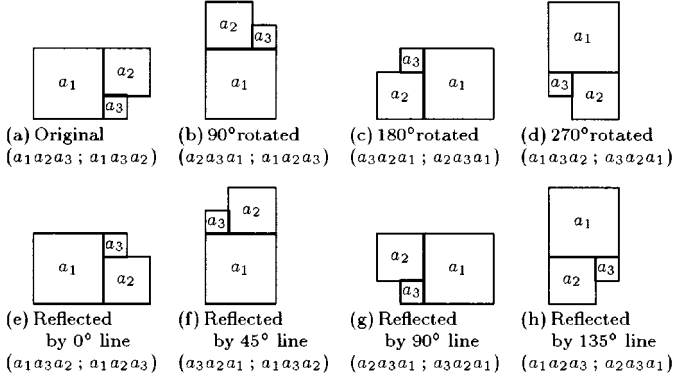
Fig. 8. Existence of positive cycle  $C$  on H-constraint graph  $G_H$ .

Fig. 9. Variation of seq-pair with rotations and reflections.

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.  $\square$

## V. CONSIDERATIONS ON THE SOLUTION SPACE

### A. Rotation and Reflection

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

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

- Op1:** Reverse the order of the former permutation (Initially,  $\Gamma_+$ ). Then all blocks are reflected by  $+45^\circ$  line.  $((a_1 a_2 a_3; a_1 a_3 a_2) \rightarrow (a_3 a_2 a_1; a_1 a_3 a_2))$ .
- Op2:** Reverse the order of the latter permutation (Initially,  $\Gamma_-$ ). Then all blocks are reflected by  $-45^\circ$  line.  $((a_1 a_2 a_3; a_1 a_3 a_2) \rightarrow (a_1 a_2 a_3; a_2 a_3 a_1))$ .
- Op3:** Exchange the former and the latter permutations ( $\Gamma_+$  and  $\Gamma_-$ ). Then all blocks are reflected by  $0^\circ$  line.  $((a_1 a_2 a_3; a_1 a_3 a_2) \rightarrow (a_1 a_3 a_2; a_1 a_2 a_3))$ .

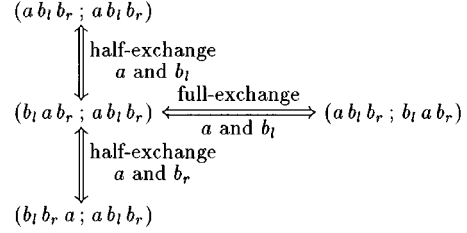
For example, the sequence of **Op1** and **Op3** makes the block rotate by  $270^\circ$  and the sequence of **Op3** and **Op1** makes the block rotate by  $90^\circ$ . The effects of the sequence of operations are displayed in Fig. 9.

### B. Reachability in the Solution Space

In [5] and [8], two kinds of adjacent seq-pairs are used for the search of simulated annealing.

TABLE I  
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

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

**Full-Exchange:** exchange of two rectangles on both  $\Gamma_+$  and  $\Gamma_-$ , and

**Half-Exchange:** exchange of two rectangles on  $\Gamma_+$ .

But these adjacent seq-pair definitions are proved easily that a certain seq-pair is not reachable from another certain seq-pair. In Table I, 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 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.

Therefore, in [20], an alternative pair of adjacent seq-pair are used:

$\Gamma_+$ -Exchange: exchange of two rectangles on  $\Gamma_+$ ;

$\Gamma_-$ -Exchange: exchange of two rectangles on  $\Gamma_-$ .

Note that  $\Gamma_+$ -exchange is the same operation as the half-exchange. If rectilinear blocks are limited to L-blocks, it is proved in [20] that any feasible seq-pairs are reachable from any feasible seq-pair with  $\Gamma_+$ - and  $\Gamma_-$ -exchanges and the diameter of the solution space is  $O(n^2)$  where  $n$  is the number of blocks. However, if a concave block is included, a pathological example, in which reachability is not preserved, exists as in Fig. 11. In this example, seq-pair is  $(a_1, a_2, b, a_3, a_4; a_2, a_4, b, a_1, a_3)$  uniquely and all of  $\Gamma_+$ ,  $\Gamma_-$ , and full-exchange cannot be applied to it. It has not been known whether reachability is preserved in the solution space or not if blocks are limited to convex ones.

## VI. 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 existence of positive cycles and to calculate the longest path length on horizontal and vertical constraint graphs.  $\Gamma_+$ -exchange and  $\Gamma_-$ -exchange, which are explained above, are used to obtain adjacent seq-pairs. The solution space is searched by simulated



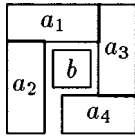


Fig. 11. A packing example of a concave block  $a$  and a rectangle block  $b$ , whose seq-pair is  $(a_1, a_2, b, a_3, a_4; a_2, a_4, b, a_1, a_3)$  uniquely and there exists no feasible adjacent seq-pair by  $\Gamma_{+-}$ ,  $\Gamma_{-}$ , and full-exchange.

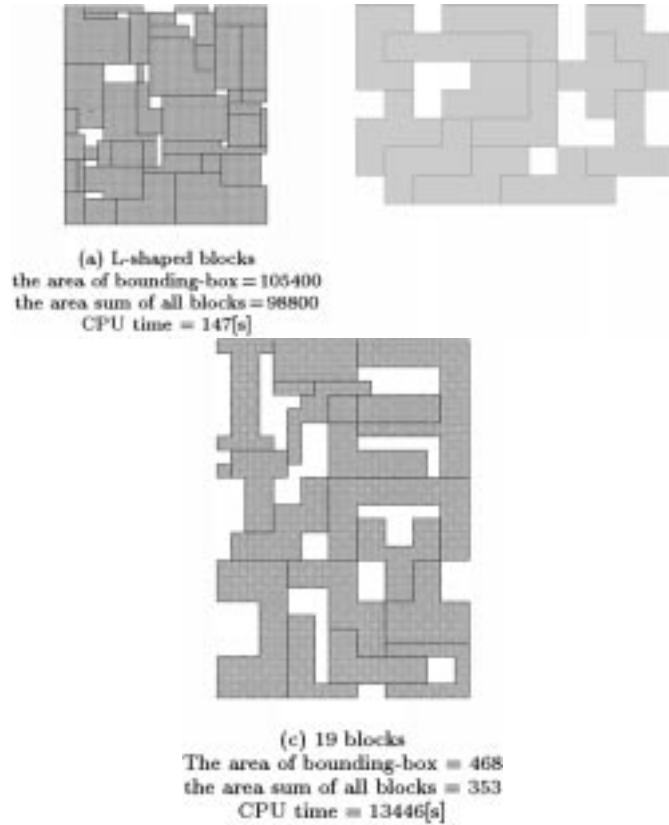


Fig. 12. Packing examples of rectilinear blocks.

annealing method. If a randomly made adjacent seq-pair is infeasible, the seq-pair is thrown away and another adjacent seq-pair is made until the seq-pair is feasible.

A packing example of the same data as Fig. 10 in [7], that consists of 10 L-blocks and 30 rectangles, is shown in Fig. 12(a). (It is reflected by X-axis.) This packing takes 147 seconds without block rotations and reflections. A packing example of 12 rectilinear blocks, which includes a concave block, is shown in Fig. 12(b). These blocks are known as “pentomino”. A packing example of 19 rectilinear blocks, which is randomly handmade, is shown in Fig. 12(c).

## VII. CONCLUSION

We have proposed a new method to represent and 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  $O(n^2 \log n)$  by applying [6] to Ford's algorithm. 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.

In the experiments, an adjacent seq-pair is made randomly until the seq-pair is feasible. As this method wastes time, a solution space that consists only of feasible seq-pairs by clever move operation, which makes an adjacent seq-pair randomly, is preferable. Another method to save time is “adaptation” technique, which is proposed in [11] to adapt each infeasible solution to a similar feasible solution. This leaves as a future work.

## ACKNOWLEDGMENT

The authors would like to thank Prof. Y. Kajitani and Prof. A. Takahashi, Tokyo Institute of Technology, for helpful comments.

## REFERENCES

- [1] Y.-Z. Liao and C. K. Wong, “An algorithm to compact a VLSI symbolic layout with mixed constraints,” *IEEE Trans. Computer-Aided Design*, vol. 2, pp. 62–69, Feb. 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*, 1985, pp. 1455–1458.
- [4] J. A. McHugh, *Algorithmic Graph Theory*. Englewood Cliffs, NJ: Prentice Hall, 1990.
- [5] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, “Rectangle-packing-based module placement,” in *Proc. ICCAD'95*, 1995, pp. 472–479.
- [6] T. Takahashi, “An algorithm for finding a maximum weight decreasing sequence in a permutation, motivated by rectangle packing problem,” *Tech. Rep. IEICE*, vol. VLD96, no. 201, pp. 31–35, 1996.
- [7] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, “Module placement on BSG structure and VLSI layout applications,” in *Proc. ICCAD'96*, 1996, pp. 484–491.
- [8] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, “VLSI module placement based on rectangle-packing by the sequence-pair,” *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 1518–1524, Dec. 1996.
- [9] 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*, 1997, pp. 265–270.
- [10] J. Dufour, R. McBride, P. Zhang, and C.-K. Cheng, “A building block placement tool,” in *Proc. ASP-DAC'97*, 1997, pp. 271–276.
- [11] H. Murata, K. Fujiyoshi, and M. Kaneko, “VLSI/PCB placement with obstacles based on sequence-pair,” in *Proc. 1997 Int. Symp. Physical Design*, Apr. 1997, pp. 26–31.
- [12] H. Murata, K. Fujiyoshi, and M. Kaneko, “VLSI/PCB placement with obstacles based on sequence-pair,” *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 60–68, Jan. 1998.
- [13] S. Nakatake, M. Furuya, and Y. Kajitani, “Module placement on BSG-structure with preplaced modules and rectilinear modules,” in *Proc. ASP-DAC'98*, 1998, pp. 571–576.
- [14] J. Xu, P.-N. Guo, and C.-K. Cheng, “Rectilinear block placement using sequence-pair,” in *Proc. Int. Symp. Physical Design*, 1998, pp. 173–178.
- [15] M. Z.-W. Kang and W. W.-M. Dai, “Topology constrained rectilinear block packing,” in *Proc. Int. Symp. Physical Design*, 1998, pp. 179–186.
- [16] K. Fujiyoshi and H. Murata, “A method to pack L-shaped and rectangular modules using sequence-pair,” in *Proc. 11-th Workshop Circuits and Systems in Karuizawa*, Apr. 1998, pp. 113–118. (in Japanese).

- [17] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module packing based on the BSG-structure and IC layout applications," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 519–530, June 1998.
- [18] M. Z. Kang and W. W.-M. Dai, "Arbitrary rectilinear block packing based on sequence pair," in *Proc. ICCAD'98*, 1998, pp. 259–266.
- [19] K. Sakanushi, S. Nakatake, and Y. Kajitani, "The multi-BSG: Stochastic approach to an optimum packing of convex-rectilinear blocks," in *Proc. ICCAD'98*, 1998, pp. 267–274.
- [20] K. Fujiyoshi and H. Murata, "A method to pack L-shaped and rectangular modules using sequence-pair" (in Japanese), *IEICE Trans. Fundamentals*, to be published.



**Kunihiro Fujiyoshi** (M'96) was born in Tokyo, in 1964. He received the B.E., M.E. and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1987, 1989, and 1994, respectively.

From 1992 to 1996, he was a Research Associate of School of Information Science at Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan. Since 1997, he has been an Associate Professor with Department of Electrical and Electronic Engineering, Tokyo University of

Agriculture and Technology, Tokyo, Japan. His research interests are in combinatorial algorithms and VLSI layout design.



**Hiroshi Murata** (S'94–A'97) was born in 1957 in Japan. He received the B.E. degree in electrical engineering from Kanazawa University, Ishikawa, Japan, in 1980. He received the M.E. and the D.E. degrees both in information science, from JAIST, Ishikawa, Japan, in 1994 and in 1997, respectively.

He was a Application Software Developer with Computer Applications Co., Ltd., Tokyo, Japan, from 1980 to 1984. He developed CAD/CAM system environment for Hybrid-IC's and PCB's, in Komatsu Murata Manufacturing Company, Ltd., Ishikawa,

Japan, from 1984 to 1997. He was a Visiting Researcher with the University of California, Berkeley, from 1997 to 1998. He has been conducting VLSI physical design tool development as the President of MicroArk Company, Ltd., Tokyo, Japan, since 1998. He also has been a Visiting Associate Professor at JAIST since 1998. His research interests are in building block style physical design algorithms.