Topology Constrained Rectilinear Block Packing for Layout Reuse

Technical Report : UCSC-CRL-97-23

Maggie Kang Wayne Dai
maggiek@cse.ucsc.edu dai@cse.ucsc.edu
Dept. of Computer Engineering
University of California, Santa Cruz

Abstract

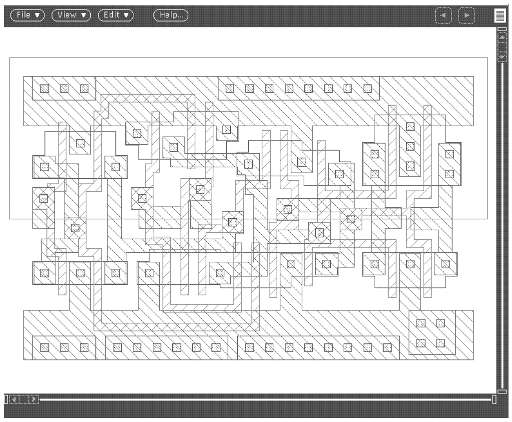
As the increasing complexity of IC design, layout reuse becomes more important. The design for renewed fabrication processes usually maintains the layout technology but using different design rules. First we extract devices and group them as a set of macro device blocks. After shrinking the macro device blocks, we would like to pack the rectilinear shaped blocks together while maintaining the original topological relationship. Such problem is referred to as *topology constrained rectilinear block packing* problem.

In this paper, we propose an efficient data representation for a special class of rectilinear polygons, called *ordered convex rectilinear polygons*, using Bounded Slicing Grid (BSG) structure. Based on both Sequence Pair (SP) and BSG structure, we propose an algorithm, which independently compacts x and y dimension under the topology constraints given the blocks are ordered convex shapes. By augumenting or further partitioning the arbitrary rectilinear polygons into the ordered convex shapes, this method can be extended to handle the general rectilinear shaped blocks.

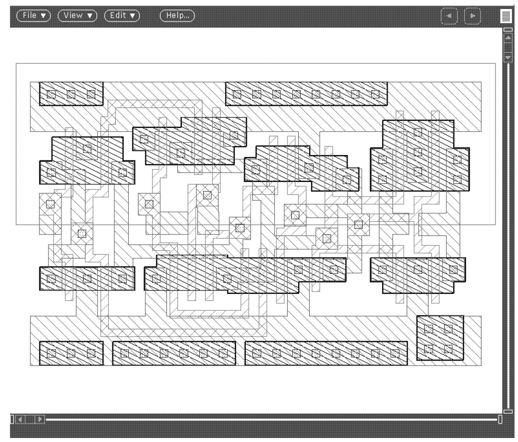
1 Block Packing in Layout Reuse

Due to the increasing complexity of IC design and short period requirement from development to market, design reuse becomes more interesting problem. Layout is one of the most complicate steps in IC design and therefore very resource consuming. Especially in the full custom layout design, various aspects of functional blocks still necessitate tedious manual work. The renewed fabrication processes usually maintain the basic layout structures but with different design rules. In order to avoid unnecessary waste of time and energy, it has become of practical importance to reuse the layout results accumulated so far in the old fabrication processes. The new design with shrunk devices and wiring can be much simplified and speeded up by taking advantage of the old design.

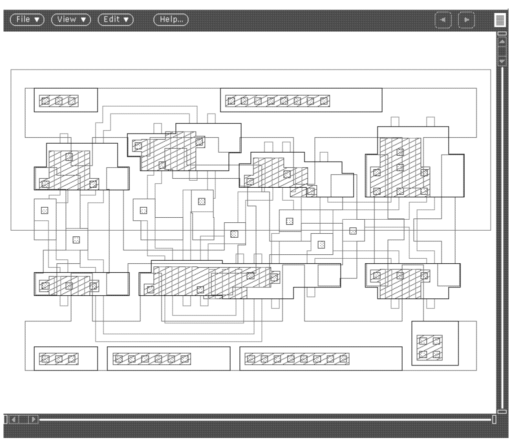
Fig. 1 illustrates our layout reuse process : the original layout is given in Fig. 1 (a), in which the devices are recognized and grouped as shown in Fig. 1 (b). Each group is called a *macro device block*, and the internal device and wiring are sized according to the new design rules. As such, the sizes of the macro device blocks are shrunk and shapes are



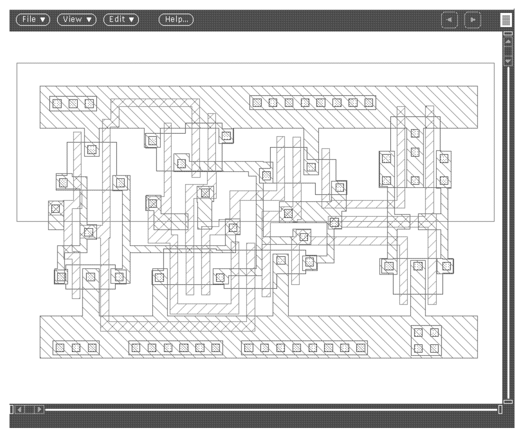
(a)



(b)



(c)



(d)

Figure 1: A practical example of the layout reuse process : the original layout is given in (a), in which the devices are recognized and grouped as shown in (b). Each group is called a *macro device block*, and the internal device and wiring are sized according to the new design rules. As such, the sizes of the macro device blocks are shrunk and shapes are changed, as shown in (c). After compacting and re-routing the inter-connection between blocks, the final layout design is achieved as shown in (d).

changed, as shown in Fig. 1 (c). After compacting the macro device blocks and re-routing the inter-connection between macro device blocks, the final layout design is achieved as shown in Fig. 1 (d).

1.1 Topology Constrained Rectilinear Packing

Given the original placement of a set of building blocks, the block sizes are shrunk and shapes are changed due to the shrunk devices and wiring in the new technology. A packing algorithm is required for compacting the sized blocks together to eliminate the empty space in between, while preserving the original topological relationship. Such packing problem, referred to as *topology constrained rectilinear block packing*, can be formulated as follows : **given a set of rectilinear-shaped blocks and their pre-placed positions, find a packing which compacts blocks together to eliminate the empty space in between, while keeping the topological relations among the blocks.** Fig. 2 shows that five blocks in the original placement are sized and compacted together. The topological relation between any two blocks is defined by their pre-placed positions. For example, block A is left to block B as shown in Fig. 2 (b). There are three key points which differentiate the new packing problem from the others:

- Packing of general rectilinear-shaped blocks;
- Preservation of topological relationship;
- Possibility of incremental update for easy incorporation of various design rules.

The last one is important for the consideration of the inter-connection between blocks. In this paper, we focus the rectilinear block packing, ignoring the interblock wiring. The incorporation of wiring will be presented in a separate paper.

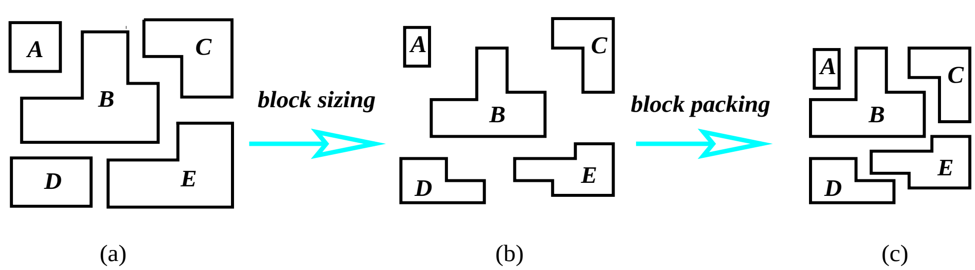


Figure 2: Given the placement of five macro blocks as shown in (a), the block sizes are shrunk and shapes are changed after the device and wire sizing, as shown in (b). In (c), they are compacted together while keeping the same topological relations as in (b).

1.2 Data Structures for Packing Problem

The interdependency of compaction in x and y dimension is the key issue for the optimal packing solution. Furthermore the topological constraints require the structure, which represents the placement of macro device blocks, clearly and accurately defines the relationship between each pair of blocks. Based on that, the local changes can be operated easily under the topological constraints. Slicing structure was proposed to represent the rectangle dissection by recursively cutting the rectangle into two parts by either horizontal or vertical line [1]. Wong and Liu proposed a normalized Polish expression to represent the slicing structure, which enables the efficient local search [2]. However, the slicing structure is very limited since most of the packings are non-slicing. To cover this intrinsic disadvantage, many efforts [3, 4, 5, 6] have been devoted but not satisfactory.

Recently, Nakatake et al. [7] introduced the bounded slicing grid structure (BSG) and Murata et al. [8] proposed the sequenced pair (SP) to represent the general rectangle packing including non-slicing structure. Both BSG and SP define the binary relationship for each pair of rectangles, and provide the way to independently compact x and y dimension. [9] applied BSG structure for the general floorplanning problem, in which the packing of L-shaped, T-shaped and soft blocks was studied. [10] indicated the complicate relationship between rectilinear blocks and proposed a SP-based compaction algorithm by using constraint graphs. Unfortunately the algorithm may leads to overlaps in the final packing, which destroys the relationship defined by SP and generates the infeasible solution.

1.3 Major Contribution of Our Work

In this paper, we propose an efficient representing method for a special class of rectilinear polygons, called *ordered convex rectilinear polygons*, in BSG structure: partitioning a rectilinear polygon into a set of sub-rectangles such that each pair of adjacent sub-rectangles form an L-shape, which fits in BSG structure very well. An algorithm is derived to independently align x and y coordinates of the sub-rectangles after BSG packing, such that the original rectilinear shape can be recovered. The related proof shows that the algorithm will not cause any overlap if every polygon has ordered convex shape, and they are partitioned and assigned into BSG structure under some constraints, which are referred to as *aligning rules*.

Based on this data representation, the topological relationship between the rectilinear polygons can be simply but accurately described using the binary relations of the corresponding sub-rectangles. As such, the topology constrained packing problem is transferred into the constrained BSG assignment problem : find out an assignment of the blocks which provides the same topological relations given by the placement, while the aligning rules are satisfied. By combining SP with BSG structure, we derive an algorithm to construct such a BSG assignment. Finally, by augumenting or further partitioning arbitrary rectilinear polygons into the ordered convex shapes, the method can be extended to general rectilinear block packing.

The rest of the paper is organized as follows : Section 2 introduces both BSG and SP structures. Section 3 describes the representing method for ordered convex rectilinear polygons in BSG structure. In Section 4, the necessary and sufficient conditions for the constrained BSG assignment are discussed. A corresponding algorithm is developed, in which the SP structure is used as a easy way to control the topological relationship. Section 5 reports the experimental results and concludes the paper.

2 Introduction of BSG and SP Structures

Nakatake et al. [7] introduced an meta-grid structure, called bounded slicing grid structure (BSG), and Murata et al. [8] proposed an equivalent structure, called sequence pair, to represent the general rectangular dissection. Both structures can provide a finite solution space at least one of which is optimal.

2.1 Bounded Slicing Grid Structure (BSG)

The BSG structure can be obtained as follows: make a row of non-overlapping horizontal line segments of two unit length and repeat them row by row, shifting by one unit length between the adjacent rows. A set of columns of vertical line segments with two unit length can be constructed in a similar way as shown in Fig. 3 (a). Those line segments are called horizontal and vertical *Bounded Slice Lines*, or BS-lines, respectively. The rectangular space surrounded by adjacent pairs of vertical and horizontal BS-lines is called *room*. BSG introduces the orthogonal relations of "right-to" and "above" to each pair of rooms uniquely. In BSG domain, a packing is represented by an assignment of rectangular blocks to rooms, called *BSG assignment*. This assignment is to map each block to a distinct room, by which the blocks inherit the relationship of the rooms.

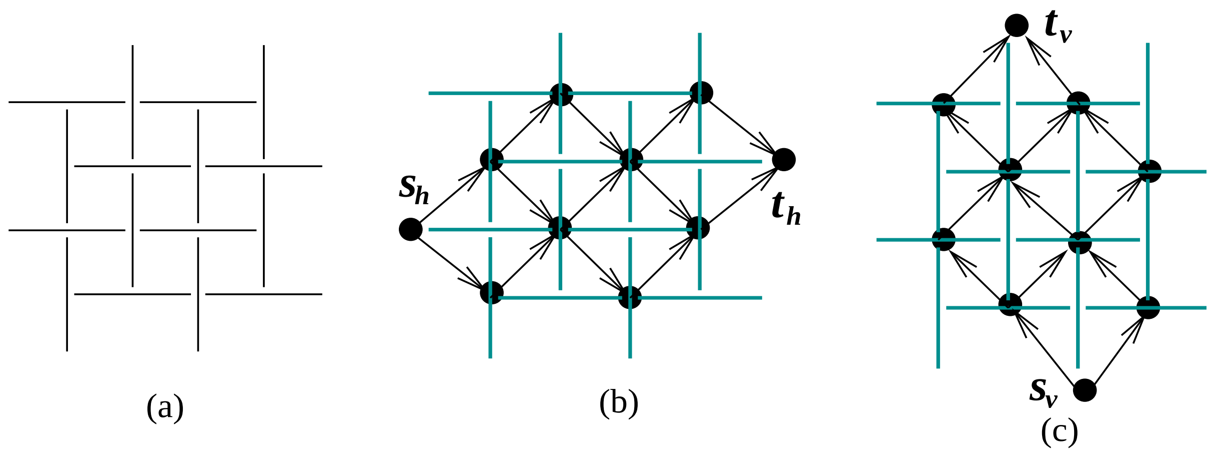


Figure 3: (a) a bounded slicing grid structure, (b) the horizontal acyclic graph G_h, and (c) the vertical acyclic graph G_v.

Two directed acyclic graphs, horizontal graph G_h and vertical graph G_v, are defined to represent the binary relations, respectively. The horizontal graph G_h puts vertex on the center of each vertical BS-line as shown in Fig. 3 (b). There is an arc from v_i to v_j if the vertical BS-line corresponding to v_j is right to the vertical BS-line corresponding to v_i and they share the same room. In particular, s_h is a source connected to all the vertices representing the leftmost BS-lines, and t_h a sink connected from all the vertices corresponding to the rightmost BS-lines. Furthermore the weight of each arc is given by the width of the block assigned to the corresponding room, if the room is occupied. Otherwise the weight is zero. The vertical graph G_v is similarly defined as shown in Fig. 3 (c).

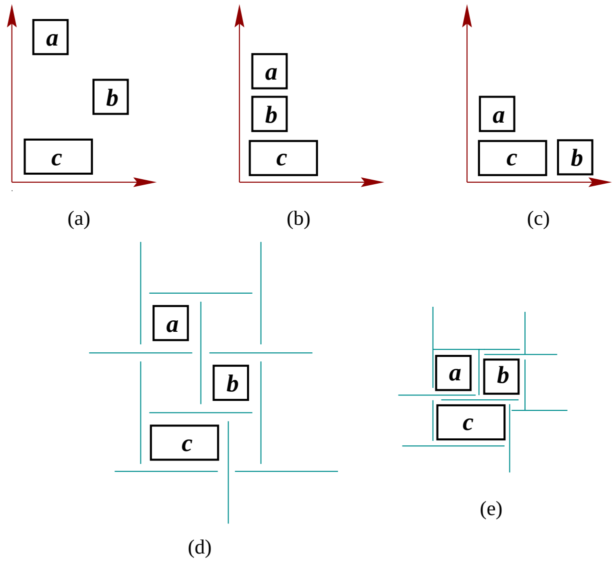
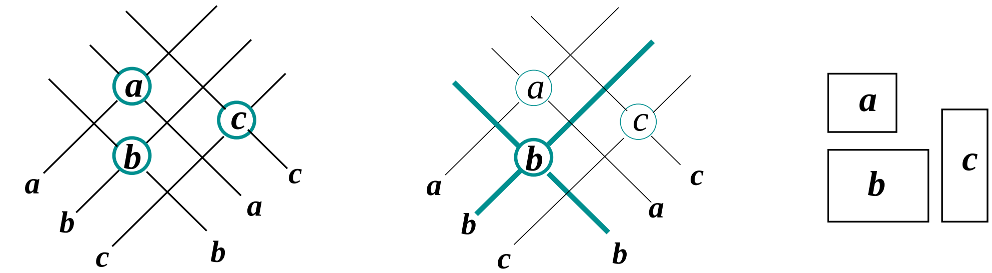


Figure 4: Given three blocks in (a), the x-y packing in (b) is achieved by first compacting x dimension followed by y dimension, the y-x packing in (c) is achieved by first compacting y dimension followed by x dimension. Neither (b) nor (c) gets the optimal solution. On the other hand, if the three blocks are assigned into BSG as in (d), the optimal packing in (e) can be achieved by independently compacting x and y dimension in BSG structure.

The x-coordinate of each block is determined by the length of the longest path from the source to the BS-line left bounding the corresponding room. In particular, the overall width equals to the length of the longest path from the source to the sink in the horizontal graph. The y-coordinate and the overall height can be determined similarly in the vertical graph. In such way, the BSG compaction is independently carried out in x and y dimension. Given three blocks as shown in Fig. 4 (a), if the compaction is first carried out in x dimension as shown in Fig. 4 (b), or the compaction is first carried out in y dimension as shown in Fig. 4 (c), the result is not optimal. On the other hand, given a BSG assignment as shown in Fig. 4 (d), the optimal compaction can be achieved by independently compacting x and y dimension as shown in Fig. 4 (e). It has been proved by [11] that there exists an assignment of n rectangular blocks in BSG domain of n rows by n columns, such that the corresponding packing is optimal.

2.2 Sequence Pair (SP)

A *sequence pair* for a set of n blocks is a pair of sequences of n symbols which represent blocks. The oblique-grid of sequence pair (abc, bac) shown in Figure 5 (a) consists of two groups of 45^0 slope lines : n slope lines of $+45^0$ are named from left to right by the symbols in the first sequence, and n slope lines of -45^0 are similarly named by the symbols in the second sequence. Each block is placed at the crossing point of the positive and negative slope lines named by the same symbol. For every block, the plane is divided by the two crossing slope lines into four cones as shown in Fig. 5 (b). Block a is in the upper cone of block b, then a is above b. Similarly, block c is in the right cone of block b, then c is right to b. In general, equivalent with BSG, SP imposes either "right-to" or

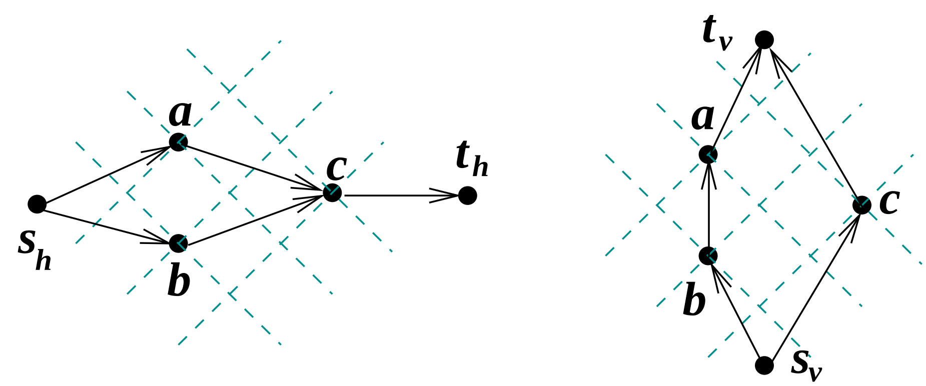


(a) (b) (c)

Figure 5: (a) the oblique-grid of sequence pair $(abc,\ bac)$, (b) the four cones of block b, and (c) the corresponding placement of a, b and c.

"above" relation for each pair of blocks :

$$(\cdots\ a \cdots b \cdots, \quad \cdots\ a \cdots b \cdots) \quad \Rightarrow \quad b \ is\ right\ to\ a,$$
$$(\cdots\ b \cdots a \cdots, \quad \cdots\ a \cdots b \cdots) \quad \Rightarrow \quad b\ is\ above\ a.$$



(a) (b)

Figure 6: The two acyclic graphs of sequence pair $(abc,\ bac)$.

Similar to BSG structure, two directed acyclic graphs can be constructed to represent the binary relationship defined by SP. In the horizontal graph G_h as shown in Fig. 6 (a), each vertex corresponds to a block, there is an arc from block a to block c if and only if block c is right to block a. In particular, there is a source s_h connected to each leftmost block and a sink t_h connected from each rightmost block. Each vertex has a weight which equals to the width of the block. The vertical graph G_v is similarly constructed as shown in Fig. 6 (b).

Given an SP of n blocks, the area minimum packing is achieved by independently compacting x and y dimension, which is equivalent to BSG compaction. It has been proved that the relations defined by every sequence pair of n rectangular blocks are satisfiable. Furthermore, there is a sequence pair which leads to the optimal packing [8].

The above introduction explores two key features for both BSG and SP : (1) the compaction of x and y dimension can be carried out independently, which is the most critical issue for the optimal packing; (2) the topological relation between each pair of rectangular blocks is uniquely defined and maintained during the compaction. Besides, it is very convenient to adjust the space between blocks by adjusting the weights in acyclic graphs without changing the topological relations. Therefore BSG and SP are both appropriate structures for the topology constrained packing problem formulated above. In the following, we will study the packing problem by focusing on BSG structure.

3 A Representing Method in BSG Structure for Ordered Convex Rectilinear Polygons

In layout reuse, the blocks can be any rectilinear shaped due to the device and wire sizing. We have studied the special cases : L-shaped and T-shaped blocks and their representation in BSG structure [9]. Intuitively L-shaped polygon was sliced into two sub-rectangles and assigned into adjacent BSG rooms. After the BSG packing, the coordinates of the sub-rectangles are aligned to recover the L-shape as shown in Fig. 7.

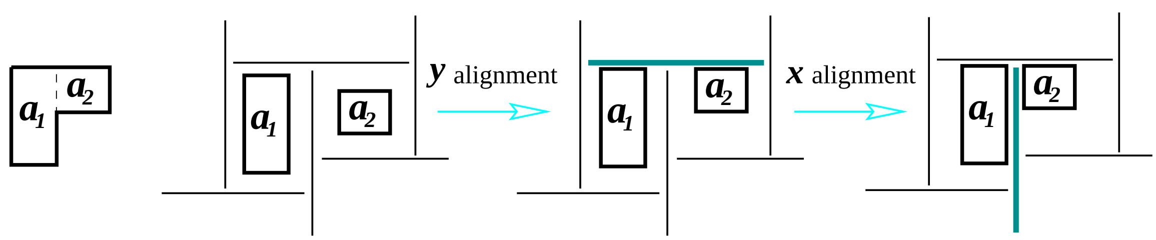


Figure 7: An L-shaped polygon in BSG structure.

For general rectilinear blocks, the similar method could be applied : partitioning a rectilinear polygon into a set of sub-rectangles, each of them is assigned to a distinct BSG room such that the coordinates of the sub-rectangles can be aligned to recover the original shape after the BSG packing. The constraints for the partition and assignment of such blocks, enforced by the alignment algorithm, are referred to as *aligning rules*. To derive the aligning rules, we first discuss the alignment method. In general, the alignment should disturb the BSG packing as little as possible : (1) the x and y coordinates are aligned independently; (2) no overlap is caused; (3) the topological relationship in BSG packing is preserved.

3.1 Coordinate Alignment of Sub-Rectangles

In BSG structure, there are two kinds of rooms : *p-typed* and *q-typed* rooms, which are located alternatively as shown in Fig. 8. Each pair of adjacent rooms are alternatively *pq*- or *qp*-adjacent. The horizontal *pq*-adjacent rooms share the bottom BS-line, and *qp*-adjacent rooms share the top BS-line.

Compared to L-shaped polygons, the alignment of more than two sub-rectangles is more complicated. Given a BSG assignment of five sub-blocks as shown in Fig. 9, a_1 and a_2 should be aligned down to BS-line u_1, while a_2 and a_3 should be aligned up to BS-line v_1, and so on. Without loss of generality, we assume the BSG compacts the blocks to the left and to the bottom. To align the y coordinate of a_1, a_2 and a_3, we may move up the BS-line u_1 such that the distance between v_1 and u_1 equals to the height of a_2 as shown in Fig. 9. In such way, a_2 can be aligned down to u_1 while up to v_1 at the same time.

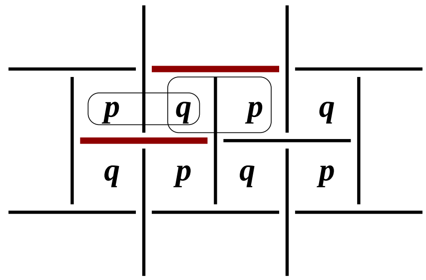


Figure 8: In BSG structure, p-typed and q-typed rooms distribute alternatively. Each pair of adjacent rooms are alternatively pq- or qp-adjacent. The horizontal pq-adjacent rooms share the bottom BS-line, and qp-adjacent rooms share the top BS-line.

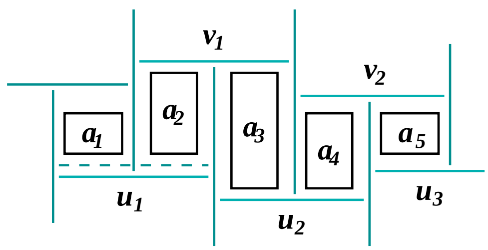


Figure 9: y alignment of sub-rectangles $A = \{a_1, a_2, \cdots, a_5\}$.

In the following, we are going to present an algorithm which aligns y coordinates of the sub-blocks assigned into the horizontally adjacent BSG rooms as shown in Fig. 9.

3.1.1 y Alignment

Given an assignment of sub-rectangles $A = \{a_1, a_2, \ldots, a_n\}$ in horizontally adjacent BSG rooms. If the room of a_1 is p-typed and a_n is q-typed as shown in Fig. 10, then n must be even : $n = 2m$, where m is an integer.

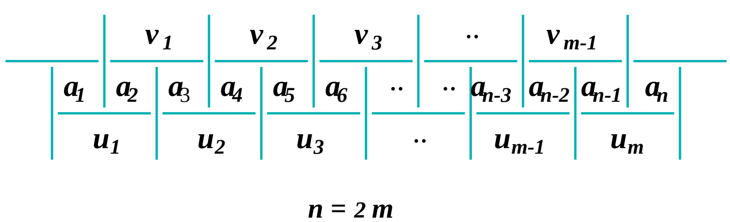


Figure 10: y alignment of sub-rectangles $A = \{a_1, a_2, \cdots, a_n\}$.

Let y_{u_i} and y_{v_i} denote the y coordinate of BS-lines u_i and v_i, respectively, and h_i denote the height of rectangle a_i. The BSG compaction in y direction has the following relations :

$$y_{v_1} = max(y_{u_1} + h_2, \quad y_{u_2} + h_3)$$
$$y_{v_2} = max(y_{u_2} + h_4, \quad y_{u_3} + h_5)$$

$$
\begin{aligned}
&\vdots \\
y_{v_{m-1}} &= max(y_{u_{m-1}} + h_{n-2}, \quad y_{u_m} + h_{n-1})
\end{aligned} \tag{1}
$$

Therefore, the y coordinate of sub-rectangles $a_1, a_2, \ldots, a_n$ can be aligned if the following relations are satisfied:

$$
\begin{aligned}
y_{u_1} + h_2 &= y_{u_2} + h_3 \\
y_{u_2} + h_4 &= y_{u_3} + h_5 \\
&\vdots \\
y_{u_{m-1}} + h_{n-2} &= y_{u_m} + h_{n-1}
\end{aligned} \tag{2}
$$

Let y'_{u_i} denote the aligned y coordinate of BS-line u_i, the non-overlapping constraint requires $y'_{u_i} \geq y_{u_i}$, that is BS-lines should never be moved downward. The aligned y coordinate of u_1 is given by :

$$
\begin{aligned}
y'_{u_1} = max(\; & y_{u_1}, \\
& y_{u_2} + h_3 - h_2, \\
& y_{u_3} + h_5 - h_4 + h_3 - h_2, \\
& \quad \vdots \\
& y_{u_{m-1}} + h_{n-3} - h_{n-4} + \cdots + h_5 - h_4 + h_3 - h_2, \\
& y_{u_m} + h_{n-1} - h_{n-2} + h_{n-3} - h_{n-4} + \cdots + h_5 - h_4 + h_3 - h_2 \;)
\end{aligned} \tag{3}
$$

Once y'_{u_1} is known, the aligned y coordinate of other BS-lines u_i, where $i > 1$, can be calculated as follows :

$$
\begin{aligned}
y'_{u_2} &= y'_{u_1} + h_2 - h_3 \\
y'_{u_3} &= y'_{u_2} + h_4 - h_5 \\
&\vdots \\
y'_{u_{m-1}} &= y'_{u_{m-2}} + h_{n-4} - h_{n-3} \\
y'_{u_m} &= y'_{u_{m-1}} + h_{n-2} - h_{n-1}
\end{aligned} \tag{4}
$$

It can be proved that for each BS-line u_i: $y'_{u_i} \geq y_{u_i}$. Therefore no overlap will occur since the horizontal BS-lines never be moved downward. For the other three cases where both room of a_1 and a_n are p-typed, or a_1 is q-typed while a_n is p-typed, or both a_1 and a_n are q-typed, the similar equations can be derived. We can conclude that the y alignment is applicable for the assignment in which the rooms of sub-rectangles are in the same row, and there is no occupied room in between. The dummy blocks with zero width can be inserted into the empty rooms in between as shown in Fig. 11. Obviously y alignment will not affect the topological relations defined by the BSG structure.

3.1.2 x Alignment

Given block $A = \{a_1, a_2, a_3, a_4, a_5\}$ assigned into the horizontally adjacent BSG rooms as shown in Fig. 12 (a), a_1 and a_2 should be aligned to the BS-line l_1 in x direction, while a_2 and a_3 should be aligned to BS-line l_2, and so on. The x coordinates can be aligned if the following condition holds:

$$
\begin{aligned}
x_{l_2} &= x_{l_1} + w_2 \\
x_{l_3} &= x_{l_2} + w_3 \\
x_{l_4} &= x_{l_3} + w_4
\end{aligned} \tag{5}
$$

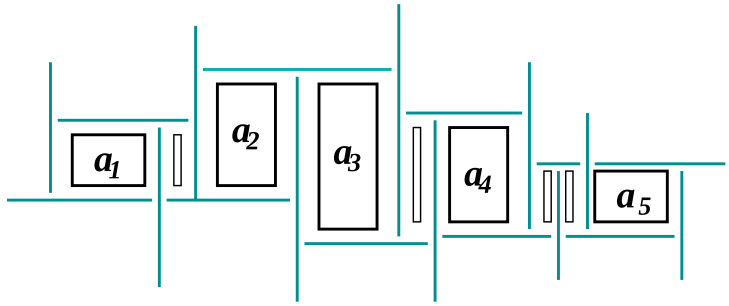


Figure 11: The BSG rooms of the sub-rectangles are in the same row, and there is no occupied room in between. Dummy blocks with zero width are inserted into the internal empty rooms, if exist.

where x_{l_i} denote the x coordinate of BS-line l_i and w_i the width of block a_i. In other words, BS-line l_3 must be exactly right to l_1 by $w_2 + w_3$. However, in the horizontal graph as shown in Fig. 12 : $x_{l_3} = max(x_{l_1} + w_2 + w_3,\ x_{l_1} + w_2' + w_2')$, where w_2' and w_3' denote the width of block a_2' and a_3', respectively. As such, the above condition may not be satisfiable. However if we move a_2 all the way to the right until hitting a_3, followed by a_1 to the right until hitting a_2 as shown in Fig. 12 (b), similarly move a_4 and a_5 to the left, the x coordinates can be aligned. No overlap is caused if the sub-rectangles satisfy :

$$h_1 \leq h_2 \leq h_3 \quad and \quad h_3 \geq h_4 \geq h_5 \tag{6}$$

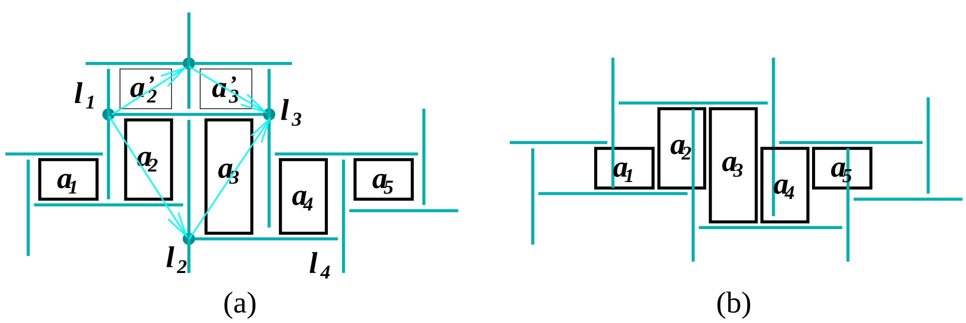


Figure 12: x alignment of sub-rectangles $\{a_1, a_2, a_3, a_4, a_5\}$.

The above property is required by x alignment. Since the blocks are moved only in horizontal direction, x alignment will not affect the vertical relations. For each right-aligned block a_i such as a_1, if any other block b is left to a_i, then b is still left to a_i after moving a_i to the right. On the other hand, $h_i \leq h_{i+1}$ according to Eq. 6. If b is right to a_i, then b is a_{i+1} itself or b is also right to a_{i+1} in the BSG packing. Thus b will be still right to a_i after the right moving of a_i. The similar situation exists for the left-aligned blocks. Therefore the topological relations of BSG packing is preserved by x alignment. Overall, the x and y coordinates are independently aligned without causing overlaps or changing the relations of BSG packing.

The symmetrical alignment method applicable for the vertically adjacent assignments of the sub-rectangles with the similar property as Eq. 6. In the following, we will derive the aligning rules which guide the block partition and assignment.

3.2 Ordered Convex Rectilinear Polygon

A rectilinear polygon A is referred to as *convex rectilinear polygon* (CRP) if and only if : given any two points inside A, there exists a shortest Manhatann path inside A. Fig. 13 (a)(1)–(6) show some convex rectilinear polygons, and (7)–(9) give three examples of non-convex rectilinear shapes.

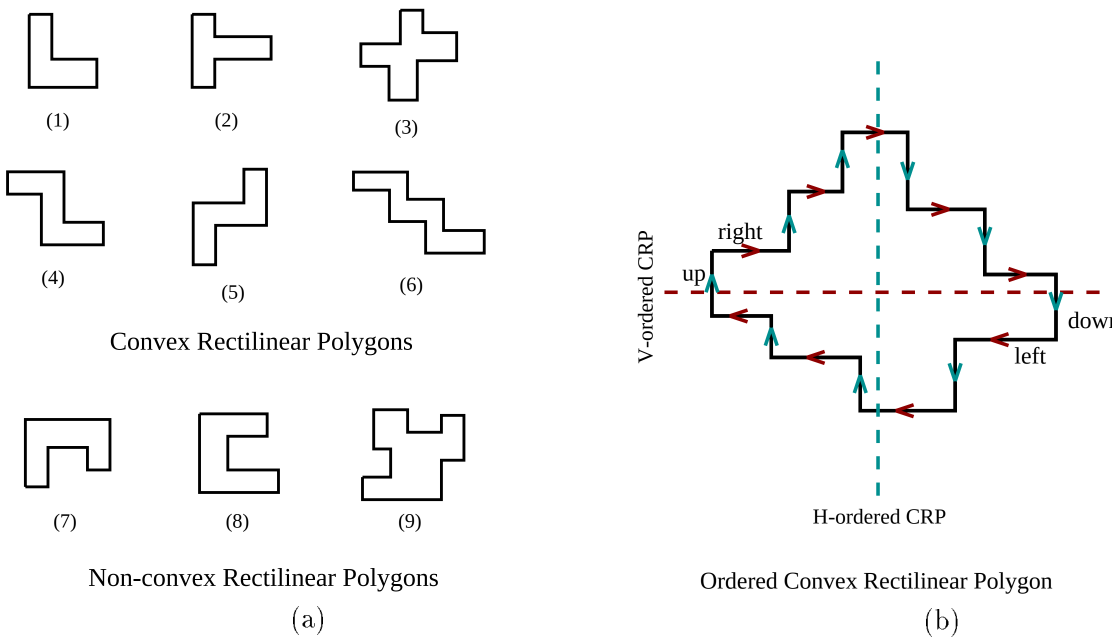


Figure 13: (a) Given a set of rectilinear polygons, in which (1)–(6) are convex shape, while (7)–(9) are non-convex. (b) When "down" edges are always right to "up" edges, such CRP is *H-ordered*. Similarly when "left" edges are always below "right" edges, such CRP is *V-ordered*.

Given a CRP A, traverse the vertices in clockwise direction and mark each edge by "up", "right", "down" and "left", respectively as shown in Fig. 13(b). A is called *H-ordered* CRP if and only if "down" edges are always right to "up" edges. Symmetrically, A is called *V-ordered* CRP if and only if "left" edges are always below "right" edges. The CRP shown in Fig. 13 (a) (1), (2) and (3) are both H-ordered and V-ordered CRP. On the other hand, the CRP shown in Fig. 13 (a) (4) is only H-ordered and Fig. 13 (a) (5) only V-ordered. However the CRP shown in Fig. 13 (a) (6) is neither H-ordered nor V-ordered.

3.2.1 Partition of Ordered CRPs

An H-ordered CRP A will be partitioned as follows :

1. Put a vertical slicing line on each vertical edge of A, the rectangular space bounded by any two adjacent slicing lines forms a sub-rectangle. In particular, the sub-rectangle bounded by two overlapped slicing lines has zero width as shown in Fig. 14 (a).
2. Visit sliced sub-rectangles from the left to right, and mark each sub-rectangle as shown in Fig. 14 (b).
3. If one sub-rectangle is marked by both p and q, bi-partition it such that the two new sub-rectangles are marked by p and q, respectively as shown in Fig. 14 (c).

We call such partition *H-partition*. Symmetrically the *V-partition* can be defined for V-ordered CRPs.

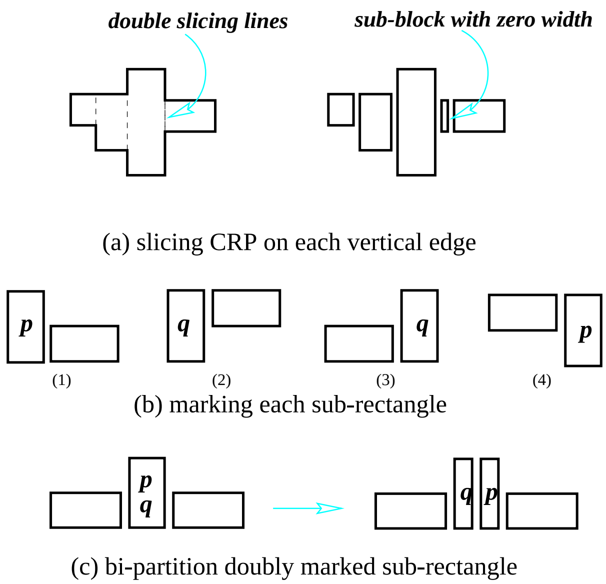


Figure 14: H-partition for an H-ordered CRP.

3.2.2 Property of Ordered CRPs

The following property of H-ordered CRP can be proved :

Lemma 1 *Given an H-ordered CRP is H-partitioned :* $A = \{a_1, a_2, \cdots, a_n\}$*, in which* a_i *is the* i^{th} *leftmost sub-rectangle, there exists a sub-rectangle* a_k*,* $k \in [1, n]$*, which is referred to as dominant sub-rectangle :*

$$h_i \le h_{i+1}, \quad \text{for } i \in [1,\ k)$$
$$h_i \ge h_{i+1}, \quad \text{for } i \in [k,\ n)$$

where h_i *denotes the height of block* a_i*.*

Similar property can be proved for V-ordered CRPs.

3.2.3 Assignment of Ordered CRPs

Given an H-partitioned CRP : $A = \{a_1,\ a_2,\ \cdots a_n\}$, in which a_i is the i^{th} left-most sub-rectangle. Let r_i denote the BSG room assigned to a_i. We call the BSG assignment of A *H-assignment* if and only if :

1. If a_i is marked by p, the room r_i is p-typed, and if a_i is marked by q, the room r_i is q-typed;
2. The room r_i is on the left of the room r_{i+1}, and they are in the same row;
3. There is no occupied room between r_i and r_{i+1}.

Similarly *V-assignment* can be defined for the V-partitioned CRP. Based on the alignment method discussed above, together with the property of Lemma 1, we can derive the following theorem:

Theorem 1 *Given a placement of a set of blocks with ordered convex rectilinear shape, the* x *and* y *dimension can be independently compacted without overlaps if each H-ordered block is H-partitioned and H-assigned, and each V-ordered block is V-partitioned and V-assigned in BSG structure.*

3.3 Constrained BSG Assignment

Given a pair of non-overlapping rectangles, there is either "right-to" or "above" relation, which is captured by BSG structure exactly. However, the topological relation between two general rectilinear polygons will be much more complicated. Rather than enumerating all possible relations as done by [10], we can simply but accurately describe such relation using the binary relations of the corresponding sub-rectangles. Given two rectilinear polygons $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3\}$ as shown in Fig. 15 (a), the relation between A and B can be defined by the relations of a_i and b_j, $i, j \in [1, 3]$ as shown in Fig. 15 (b), which are illustrated by the relation diagram shown in Fig. 15 (c).

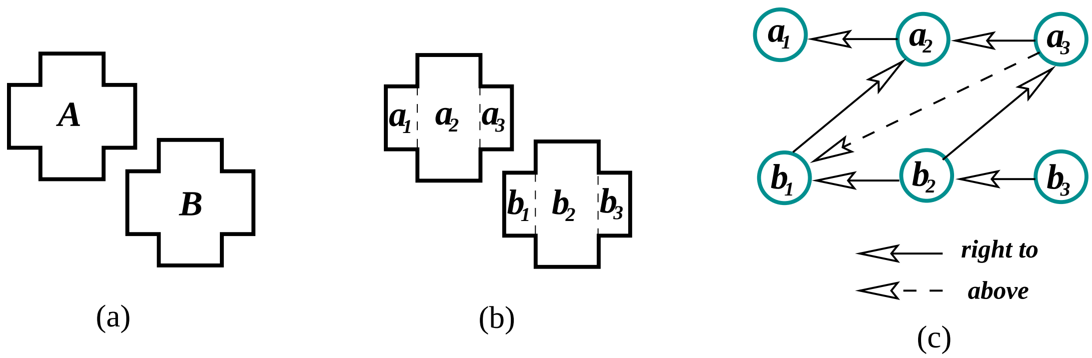


Figure 15: Given two rectilinear polygons A and B in (a), the topological relations can be described using the binary relations of the corresponding sub-blocks in (b), which are illustrated by the relation diagram in (c).

If a sub-rectangle of B is right to a sub-rectangle of A, we say B is right to A. Similarly we can define B below A. Due to the general rectilinear shape, there may exist multiple relations between two polygons. We call A and B have *consistent relationship* if and only if B is not both right to and left to A, and B is not both above and below A.

Lemma 2 *Any two convex rectilinear polygons have the consistent relationship.*

Based on the data representation presented above, the topology constrained rectilinear block packing can be transferred to a constrained BSG assignment problem : given a set of rectilinear blocks with ordered convex shapes, in which H-ordered CRPs are H-partitioned and V-ordered CRPs are V-partitioned, find a BSG assignment in which the H-partitioned CRPs are H-assigned and V-partitioned CRPs are V-assigned, while the topological relations defined in the BSG structure are the same with the given placement. In the following, we will propose an algorithm to construct such a BSG assignment for a given placement.

4 Constrained BSG Assignment

To construct such a BSG assignment, we decompose the problem into two steps : (1) construct a BSG assignment which provides the equivalent relations with the given placement; (2) each H-partitioned CRP is H-assigned and V-partitioned CRP is V-assigned. As introduced earlier, SP defines the binary relation between each pair of blocks by the order of their symbols in both sequences. Given n rectangular blocks and their topological relations, a sequence pair can be easily constructed in $O(n^2)$ time [8]. In the following, we state a method proposed by S.

Nakatake and K. Fujiyoshi, which constructs a BSG assignment for a given SP such that they defines the exact same topological relationship.

4.1 SP-based BSG Assignment

Here we adopt a coordinate system composed by two sets of $+45^0$ and -45^0 slant integer axes, both ordered from the left side as shown in Fig. 16 (a). A room centered at the cross of i_+^{th} positive and i_-^{th} negative axes is referred to by $r(i_+, i_-)$:

Fact 1 *In the slant coordinate system, if $r(0,0)$ is assumed to be a p-typed BSG room, then $r(i_+,\ i_-)$ is a p-typed room if and only if both i_+ and i_- are even. On the other hand, $r(i_+,\ i_-)$ is a q-typed room if and only if both i_+ and i_- are odd.*

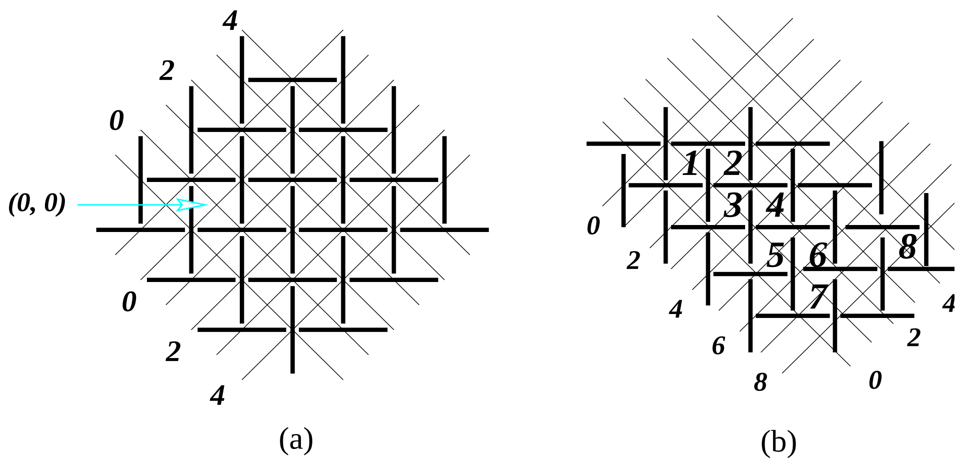


Figure 16: (a) The slant coordinate system of BSG structure, (b) the BSG assignment for the given sequence pair : $\Gamma_+ = a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8$ and $\Gamma_- = a_1\ a_3\ a_5\ a_7\ a_6\ a_4\ a_2\ a_8$.

Let (Γ_+, Γ_-) denote the given sequence pair, and $\Gamma_+(a_i)$ denote the index of block a_i in the first sequence Γ_+. Without loss of generality, we assume the first sequence $\Gamma_+ = a_1\ a_2\ \cdots\ a_n$, by relabeling if necessary, so $\Gamma_+(a_i) = i$ for $i \in [1, n]$. For example, $\Gamma_+ = a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8$, and $\Gamma_- = a_1\ a_3\ a_5\ a_7\ a_6\ a_4\ a_2\ a_8$. The SP-based BSG assignment can be constructed as follows:

1. Placing a dummy block a_0 at the beginning of Γ_- : $a_0\ a_1\ a_3\ a_5\ a_7\ a_6\ a_4\ a_2\ a_8$, and assigning $\Gamma_+(a_0) = 0$.
2. Traversing Γ_- from left to right and grouping every maximal sub-sequence which is either consecutive blocks whose $\Gamma_+()$ values are even and decreasing, or consecutive blocks whose $\Gamma_+()$ values are odd and increasing. In the above example, $\Gamma_- = [a_0]\,[a_1\ a_3\ a_5\ a_7]\,[a_6\ a_4\ a_2]\,[a_8]$. A grouped sub-sequence is called a *group*. The $\Gamma_+()$ values of blocks in a group are uniquely even or odd, thus the group is called *even* or *odd* accordingly. For example, $[a_1\ a_3\ a_5\ a_7]$ is an odd group, and $[a_6\ a_4\ a_2]$ is a even group.
3. Placing an empty group between every pair of consecutive even groups or consecutive odd groups : $[a_0]\ [a_1\ a_3\ a_5\ a_7]\ [a_6\ a_4\ a_2]\ [\,]\ [a_8]$.
4. $\Gamma_-(a_i)$ denotes the number of groups in Γ_- before the group that contains a_i. In this example, $\Gamma_-(a_1) = 1$ and $\Gamma_-(a_8) = 4$.

5. Assigning block a_i into BSG room $r(\Gamma_+(a_i),\ \Gamma_-(a_i))$: a_1 will be assigned into room $r(1,1)$ while a_8 to room $r(8,4)$, as shown in Fig. 16 (b).

The following property can be proved :

Lemma 3 *In SP-based BSG assignment, each cross $r(\Gamma_+(a_i),\ \Gamma_-(a_i))$ is a BSG room, and the relation between each pair of rooms $r(\Gamma_+(a_i),\ \Gamma_-(a_i))$ and $r(\Gamma_+(a_j),\ \Gamma_-(a_j))$ is exactly the same relation between the corresponding blocks a_i and a_j defined in the given SP.*

Using this method, a BSG assignment of n blocks can be constructed such that it provides the same relations with the given placement. To incorporate the H-assignment and V-assignment into the construction, we first derive the necessary and sufficient conditions for such assignments. Since H-assignment and V-assignment are symmetrical, we will only focus on H-assignment.

4.2 Necessary and Sufficient Conditions for H-Assignment

Lemma 4 *In the SP-based assignment, an H-partitioned CRP $A = \{a_1,\ a_2,\ \cdots,\ a_n\}$ is H-assigned if and only if :*

1. *If a_i is p-marked, both $\Gamma_+(a_i)$ and $\Gamma_-(a_i)$ should be even; if a_i is q-marked, both $\Gamma_+(a_i)$ and $\Gamma_-(a_i)$ should be odd.*
2. *If a_i and a_j are adjacent sub-blocks in A, and a_i is left to a_j, then $\Gamma_+(a_j) - \Gamma_+(a_i) = \Gamma_-(a_j) - \Gamma_-(a_i)$.*

Due to the Fact 1, the first condition above is equivalent to the first requirement of H-assignment defined in Section 3.2.3. In the slant coordinate system, room $r(i_+, i_-)$ and $r(j_+, j_-)$ are in the same row if and only if $j_+ - i_+ = j_- - i_-$. Therefore the second condition above is equivalent to the second requirement of H-assignment. As such, both conditions are necessary for H-assignment. On the other hand, if there is an occupied room $r(\Gamma_+(a_k), \Gamma_-(a_k))$ between a_i and a_j :

$$\Gamma_+(a_i) < \Gamma_+(a_k) < \Gamma_+(a_j), \qquad \Gamma_-(a_i) < \Gamma_-(a_k) < \Gamma_-(a_j)$$

then both sequences should be like : $a_i\ \cdot\cdot\ a_k\ \cdot\cdot\ a_j$, which implies that block a_k is right to a_i and left to a_j. If a_k belongs to the same CRP with a_i and a_j, then a_k must be between a_i and a_j, which conflicts to the assumption that a_i and a_j are adjacent. On the other hand, if a_k belongs to a distinct CRP, this CRP will be both left to and right to the CRP of a_i and a_j, which conflicts to the consistent relationship in Lemma 2. Therefore the rooms between a_i and a_j can not be occupied and the third requirement of H-assignment in Section 3.2.3 will be automatically satisfied in the SP-based assignment. As such, the above two conditions are sufficient for H-assignment. In the following, we will propose two operations on SP such that the SP-based assignment satisfies the two conditions of Lemma 4.

4.3 PQ-Adjustment

To satisfy the first condition of Lemma 4, we define an operation called *pq-adjustment*. In the SP-based assignment, $\Gamma_+(i)$ and $\Gamma_-(i)$ are both even or both odd. Without loss of generality, we assume a_i is a p-marked block, $\Gamma_+(i)$ and $\Gamma_-(i)$ are both odd. pq-adjustment is carried out by inserting two dummy blocks * into the first sequence Γ_+, one right before and the other right after a_i, respectively, and appending two empty groups at the end of the second sequence Γ_-, as shown in Fig. 17 (a).

After this operation, $\Gamma_+(i)$ is increased by one and becomes even. The $\Gamma_+()$ values of those blocks after a_i in the first sequence are increased by two. The parity of $\Gamma_+()$ values will not be affected except block a_i. Given a_j is the predecessor of a_i in the second sequence, overall there are four possible cases as shown in Fig. 17 (b) :

1. $\Gamma_+(j)$ is odd, a_j and a_i are originally grouped together as shown in Fig. 17 (b) (1). The group will split when $\Gamma_+(i)$ becomes even after the operation. So the number of groups between a_j and a_i is increased by one.
2. $\Gamma_+(j)$ is odd, a_j and a_i are grouped separately, an empty group must be in between as shown in Fig. 17 (b) (2). When $\Gamma_+(i)$ becomes even, the empty group is deleted, and the number of groups between a_j and a_i is decreased by one.
3. $\Gamma_+(j)$ is even, a_j and a_i are grouped separately, as shown in Fig. 17 (b) (3). When $\Gamma_+(i)$ becomes even, which is greater than $\Gamma_+(j)$, a_j and a_i will be grouped separately and one empty group is inserted in between, as shown in Fig. 17 (b) (3). The number of groups between a_j and a_i is increased by one.
4. $\Gamma_+(j)$ is even, a_j and a_i are grouped separately, as shown in Fig. 17 (b) (4). When $\Gamma_+(i)$ becomes even, which is smaller than $\Gamma_+(j)$, a_j and a_i will be grouped together, and the number of groups between a_j and a_i is reduced by one.

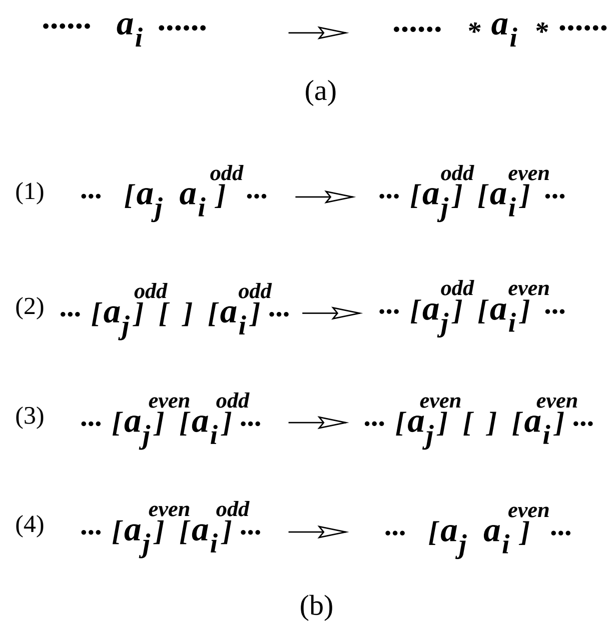


Figure 17: pq-adjustment inserts two dummy blocks in the first sequence, right before and after a_i, respectively as shown in (a). $\Gamma_+(a_i)$ is increased by one, and the $\Gamma_+()$ values of blocks after a_i will be increased by two. On the other hand, given a_j is the predecessor of a_i in the second sequence, overall there are four possible cases as shown in (b) (1) – (4). It can be derived that the number of groups between a_j and a_i will be increased or decreased by one.

Since the parity of $\Gamma_+()$ values are preserved for the other blocks, the groups before a_j in the second sequence will not be changed, and $\Gamma_-()$ values remain the same for those blocks before a_i in the second sequence. Due to the changed groups between a_j and a_i, $\Gamma_-(a_i)$ is increased or decreased by one and becomes even.

On the other hand, given a_k is the successor of a_i in the second sequence, the similar analysis derives that the number of groups between a_i and a_k will be increased or decreased by one, and the groups after a_k in the second sequence will not be affected. So together with the changed groups between a_j and a_i, the $\Gamma_-()$ values are changed by either 0 or 2 for those blocks after a_i in the second sequence. Overall we can conclude :

Lemma 5 *Given block a_i is p-marked, and the corresponding room $(\Gamma_+(a_i),\ \Gamma_-(a_i))$ is q-typed, pq-adjustment can adjust the room of a_i to p-typed by simultaneously changing the parity of $\Gamma_+(a_i)$ and $\Gamma_-(a_i)$. Furthermore the pq-adjustment carried for block a_i will not affect the parity of $\Gamma_+()$ or $\Gamma_-()$ values of the other blocks.*

Similarly, the pq-adjustment can be applied for q-marked block. In such way, the first condition of Lemma 4 can be satisfied for all marked blocks by carrying out pq-adjustment for each of them, individually.

4.4 Δ-Adjustment

Given sub-blocks a_i and a_j are adjacent and a_i is left to a_j, both sequences should be : $a_i \cdot\cdot a_j$. We define Δ_+^{ij} and Δ_-^{ij} as follows :

$$\Delta_+^{ij} = \Gamma_+(a_j) - \Gamma_+(a_i), \qquad \Delta_-^{ij} = \Gamma_-(a_j) - \Gamma_-(a_i).$$

The second condition of Lemma 4 is equivalent to $\Delta_+^{ij} = \Delta_-^{ij}$. The following Lemma can be proved :

Lemma 6 $\mid \Delta_+^{ij} - \Delta_-^{ij} \mid = 2m$*, where m is an integer.*

When $\Delta_-^{ij} - \Delta_+^{ij} = 2m > 0$, an operation called Γ_+-*adjustment* is applied by consecutively inserting $2m$ dummy blocks * in the first sequence Γ_+, somewhere between a_i and a_j (the exact position will be discussed later), while appending $2m$ empty groups at the end of the second sequence Γ_-.

In the example shown in Fig. 18, given a_1 and a_3 are adjacent sub-blocks, $\Delta_-^{13} - \Delta_+^{13} = 4$. Four dummy blocks are inserted in the first sequence while four empty groups attached at the end of the second sequence. Obviously, $\Gamma_+(3)$ is increased by four and accordingly Δ_+^{13} is increased by four. On the other hand, the parity of the $\Gamma_+()$ values are not affected due to the even number of dummy blocks, so the groups of the second sequence will not be affected, and Δ_-^{ij} remains the same. Therefore $\Delta_+^{13} = \Delta_-^{13}$ after the Γ_+-adjustment.

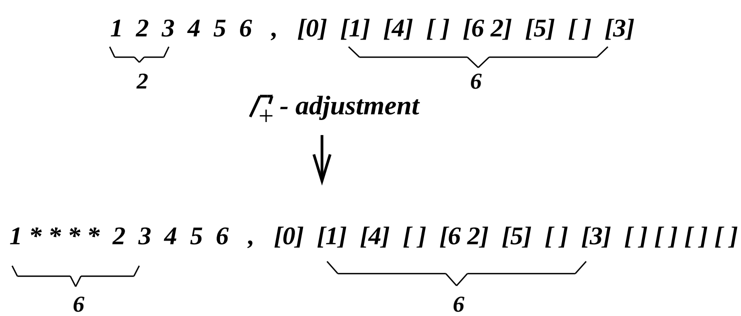


Figure 18: Given a sequence pair and adjacent sub-blocks a_1 and a_3 : $\Delta_-^{13} - \Delta_+^{13} = 4$. Γ_+-adjustment is applied: insert four dummy blocks $*$ into the first sequence between a_1 and a_3, while attach four empty groups [ ] at the end of the second sequence. As such, $\Delta_+^{13} = \Delta_-^{13}$.

Similarly when $\Delta_+^{ij} - \Delta_-^{ij} = 2m > 0$, another operation called Γ_--*adjustment* is applied : consecutively inserting $2m$ empty groups [ ] in the second sequence Γ_-, somewhere between the groups contain a_i and a_j (the exact position will be discussed later). If a_i and a_j are originally grouped together, the group will split and $2m$ empty groups are inserted in between. On the other hand, $2m$ dummy blocks $*$ are appended at the end of the first sequence Γ_+. So the $\Gamma_+()$ values remain the same, while $\Gamma_-()$ values of those blocks after the empty groups are increased by $2m$. Therefore Δ_-^{ij} is increased by $2m$ and $\Delta_+^{ij} = \Delta_-^{ij}$. Overall we call both operations *Δ-adjustment*.

4.4.1 Two Basic Properties of SP

Given two pairs of adjacent blocks (a_i, a_j) and (b_i, b_j), their relative order in a sequence will be one of the following three cases :

- $a_i \cdot\cdot a_j \cdot\cdot b_i \cdot\cdot b_j \Rightarrow$ a-pair *separates* from b-pair;
- $a_i \cdot\cdot b_i \cdot\cdot a_j \cdot\cdot b_j \Rightarrow$ a-pair *interleaves* with b-pair;
- $a_i \cdot\cdot b_i \cdot\cdot b_j \cdot\cdot a_j \Rightarrow$ a-pair *includes* b-pair.

The following two properties can be proved :

Lemma 7 *If a-pair includes b-pair in one sequence of SP, then a-pair separates from b-pair in the other sequence.*

Lemma 8 *If a-pair interleaves with b-pair in one sequence of SP, then a-pair separates from b-pair in the other sequence.*

Since the proofs of the above two Lemmas are very similar, we will only show the first one. Without loss of generality, we assume a_i is left to a_j, and b_i left to b_j. Then both Γ_+ and Γ_- will have : $a_i \cdot\cdot a_j$ and $b_i \cdot\cdot b_j$. If a-pair includes b-pair in the first sequence : $\Gamma_+ = a_i \cdot\cdot b_i \cdot\cdot b_j \cdot\cdot a_j$, then in the second sequence b_i will not be between a_i and a_j. Otherwise, $(a_i\ b_i\ a_j,\ a_i\ b_i\ a_j)$ implies b_i is right to a_i and left to a_j. With this relationship, if b_i belongs to the same CRP with a_i and a_j, b_i will be left to a_i while right to a_j, which conflicts to the assumption that a_i and a_j are adjacent. On the other hand, if b_i belongs to a distinct CRP, the CRP of b_i will be both left to and right to the CRP of a_i and a_j, which conflicts to the consistent relationship of Lemma 2. Therefore, the second sequence must be either $b_i \cdot\cdot a_i \cdot\cdot a_j$ or $a_i \cdot\cdot a_j \cdot\cdot b_i$. The same situation happens for b_j. Thus there are only three possible permutations for the second sequence Γ_-:

$$b_i \cdot\cdot a_i \cdot\cdot a_j \cdot\cdot b_j$$
$$a_i \cdot\cdot a_j \cdot\cdot b_i \cdot\cdot b_j$$
$$b_i \cdot\cdot b_j \cdot\cdot a_i \cdot\cdot a_j$$

If Γ_- is in the first case, we can derive the relation graph as shown in Fig. 19 (a). If a_i and a_j are adjacent sub-blocks as shown in Fig. 19 (b), b_i and b_j should be located at the two shadowed cones, respectively. So they could not be adjacent. Similarly a_i and a_j could not be adjacent given b_i and b_j are adjacent as shown in Fig. 19 (c). Therefore we can conclude Γ_- can only be one of the last two cases, in which a-pair separates from b-pair.

4.4.2 Δ-Adjustment Ordering

To satisfy the second condition of Lemma 4, Δ-adjustment will be carried out individually for each pair of adjacent sub-blocks. It can be proved that :

Lemma 9 *Given a non-overlapping placement of n blocks, there exists a SP such that all adjacent sub-blocks can be ordered such that :*

1. If a-pair includes b-pair in one sequence : $a_i\ b_i\ b_j\ a_j$, b-pair will be operated earlier than a-pair;

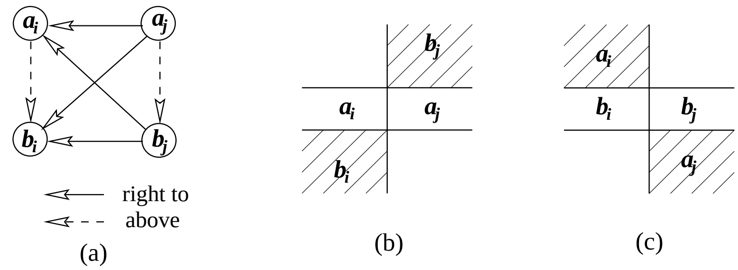


Figure 19: If the relations between blocks a_i, a_j, b_i, and b_j are as shown in (a), assume a_i and a_j are adjacent sub-blocks as shown in (b), the b_i and b_j would be located at the two shadowed cones, respectively. They could not be adjacent. If we assume b_i and b_j are adjacent sub-blocks as shown in (c), a_i and a_j could not be adjacent, either.

2. If a-pair, b-pair and c-pair interleave with each other in one sequence : $a_i \cdot\cdot b_i \cdot\cdot c_i \cdot\cdot a_j \cdot\cdot b_j \cdot\cdot c_j$, b-pair will be operated earlier than a-pair or c-pair.

Given two adjacent sub-blocks a_i and a_j, when Δ-adjustment is carried out for a-pair, $2m$ dummy blocks or empty groups are inserted in one sequence, the inserting position is within some range between a_i and a_j, denoted by $I(a_i, a_j)$:

1. If b-pair interleaves with a-pair as : $b_i \cdot\cdot a_i \cdot\cdot b_j \cdot\cdot a_j$, and the ordering of b-pair is earlier than the ordering of a-pair, then $I(a_i, a_j)$ must be right to b_j;
2. If c-pair interleaves with a-pair as : $a_i \cdot\cdot c_i \cdot\cdot a_j \cdot\cdot c_j$, and the ordering of c-pair is earlier than the ordering of a-pair, then $I(a_i, a_j)$ must be left to c_i.

The following Lemma can be derived :

Lemma 10 *Given the adjacent sub-blocks are ordered according to Lemma* 9*, the Δ-adjustment are carried out in this order for each pair of adjacent sub-blocks, the second condition of Lemma* 4 *will be satisfied.*

Since Δ-adjustment doesn't change the parity of $\Gamma_+()$ or $\Gamma_-()$ values, the first condition of Lemma 4 will not be affected. Overall we can conclude the following theorem :

Theorem 2 *The necessary and sufficient conditions for H-assignment in Lemma* 4 *can be satisfied by applying pq-adjustment and Δ-adjustment in SP-based BSG assignment.*

The same operations can also be applied to SP-based BSG assignment such that the necessary and sufficient conditions for V-assignment are satisfied.

4.5 One Example of Constrained BSG Assignment

In the following, we will give an example to show how the *pq*-adjustment and Δ-adjustment are carried out in the SP-based BSG assignment. Given the placement of five blocks as shown in Fig. 20, in which four L-shaped blocks are either H-partitioned or V-partitioned. The sequence pair extracted from the placement is as follows :

$$\Gamma_+ = a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9, \quad \Gamma_- = a_1\ a_8\ a_4\ a_7\ a_5\ a_3\ a_6 a_2\ a_9.$$

To simplify the notation, we abbreviate a_i as i, and let $*^n$ and $[\ ]^n$ denote the n consecutive dummy blocks and empty groups, respectively. In addition, we ignore the dummy blocks or empty groups attached at the end of the sequences. The grouped SP is as follows :

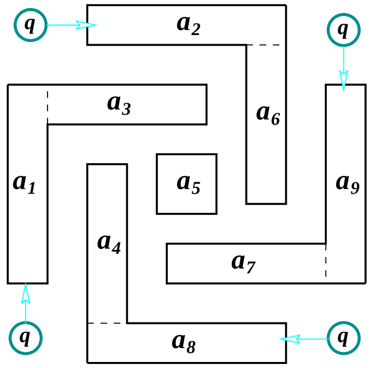


Figure 20: Given the placement of five blocks, in which four L-shaped blocks are either H-partitioned or V-partitioned.

$$(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9, \quad [0]\ [1]\ [8\ 4]\ [7]\ [\ ]\ [5]\ [\ ]\ [3]\ [6\ 2]\ [9]).$$

Blocks a_1, a_2, a_8 and a_9 are q-marked, in which the first condition of Lemma 4 has already been satisfied for blocks a_1 and a_9, so the pq-adjustment is carried out only for a_2 and a_8 :

$$(1\ \underline{\ast}\ 2\ \underline{\ast}\ 3\ 4\ 5\ 6\ 7\ \underline{\ast}\ 8\ \underline{\ast}\ 9, \quad [0]\ \underline{[1\ 8]\ [4]}\ [7]\ [\ ]\ [5]\ [\ ]\ [3]\ \underline{[6]\ [2\ 9]}).$$

There are totally four pairs of adjacent sub-blocks, and according to Lemma 9 they are ordered as follows:

$$(a_4,\ a_8) > (a_2,\ a_6) > (a_1,\ a_3) > (a_7,\ a_9),$$

in which $>$ means "earlier than". Δ-adjustment is carried out for each of them in such order. First four empty groups are inserted into Γ_- for pair (a_4, a_8):

$$(1\ \ast\ 2\ \ast\ 3\ 4\ 5\ 6\ 7\ \ast\ 8\ \ast\ 9, \quad [0]\ [1\ 8]\ \underline{[\ ]^4}\ [4]\ [7]\ [\ ]\ [5]\ [\ ]\ [3]\ [6]\ [2\ 9]).$$

Second four empty groups are inserted into Γ_- for pair (a_2, a_6):

$$(1\ \ast\ 2\ \ast\ 3\ 4\ 5\ 6\ 7\ \ast\ 8\ \ast\ 9, \quad [0]\ [1\ 8]\ [\ ]^4\ [4]\ [7]\ [\ ]\ [5]\ [\ ]\ [3]\ [6]\ \underline{[\ ]^4}\ [2\ 9]).$$

Then six dummy blocks are inserted into Γ_+ for pair (a_1, a_3):

$$(1\ \ast\ \underline{\ast^6}\ 2\ \ast\ 3\ 4\ 5\ 6\ 7\ \ast\ 8\ \ast\ 9, \quad [0]\ [1\ 8]\ [\ ]^4\ [4]\ [7]\ [\ ]\ [5]\ [\ ]\ [3]\ [6]\ [\ ]^4\ [2\ 9]).$$

At last, six dummy blocks are inserted into Γ_+ for pair (a_7, a_9):

$$(1\ \ast^7\ 2\ \ast\ 3\ 4\ 5\ 6\ 7\ \ast\ 8\ \ast\ \underline{\ast^6}\ 9, \quad [0]\ [1\ 8]\ [\ ]^4\ [4]\ [7]\ [\ ]\ [5]\ [\ ]\ [3]\ [6]\ [\ ]^4\ [2\ 9]).$$

Finally the SP-based assignment is as follows:

$$\begin{array}{lll} a_1 \to (1,\ 1) & a_2 \to (9,\ 17) & a_3 \to (11,\ 11) \\ a_4 \to (12,\ 6) & a_5 \to (13,\ 9) & a_6 \to (14,\ 12) \\ a_7 \to (15,\ 7) & a_8 \to (17,\ 1) & a_9 \to (25,\ 17) \end{array}$$

The necessary and sufficient conditions of both H-assignment and V-assignment are satisfied.

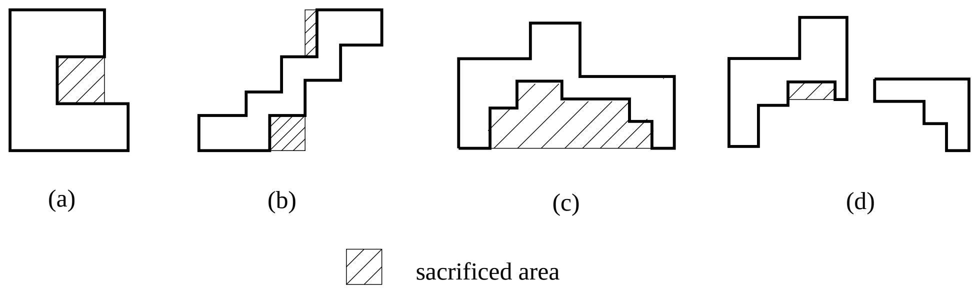


Figure 21: A rectilinear polygon can be transferred to an ordered convex shape by sacrificing some area, as shown in (a) and (b). However, the sacrificed area may be too large to be ignored as shown in (c). Thus a further partition of the original macro block is appropriate as shown in (d).

5 Experimental Results and Conclusion

For the application of the layout reuse problem, the constraint of ordered convex shape may be too restrict. Ideally any rectilinear shaped block can be transferred to an ordered convex shape by sacrificing a minimum area as shown in Fig. 21 (a) and (b). After the packing, those sacrificed area can be utilized as the routing area. In some cases as shown in Fig. 21 (c), the sacrificed area may be even larger than the area of original block. A further partition of the macro block into a set of ordered convex sub-blocks as shown in Fig. 21 (d) is more appropriate, which may requires the knowledge about the layout structure inside the macro block.

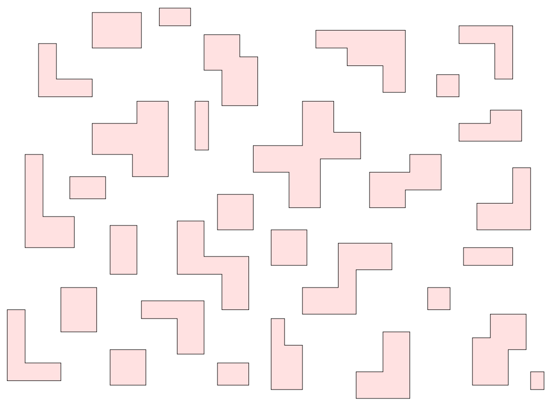
As such, how to transfer a general rectilinear block into an ordered convex shaped block with minimum additional area, and how to partition a rectilinear block into a minimum number of ordered convex shaped sub-blocks become interesting problems. We will not present the algorithms here due to the limitation of the paper length.

5.1 Experimental Results

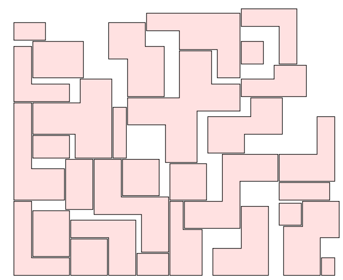
To demonstrate the efficiency of the algorithm presented in this paper, we randomly generated the example shown in Fig. 22 (a), in which all of 31 blocks have ordered convex rectilinear shapes. The packing result achieved by our algoirithm is shown in Fig. 22 (b), in which the x and y dimension are independently compacted and the topological relations of blocks in (a) are preserved. On the other hand, we compact the 31 blocks without considering the relation constraints, the packing result shown in Fig. 22 (c) is achieved by first packing x dimension followed by y dimension, and Fig. 22 (d) is the result by first packing y dimension followed by x dimension. Obviously, our algorithm give the best result.

5.2 Conclusion

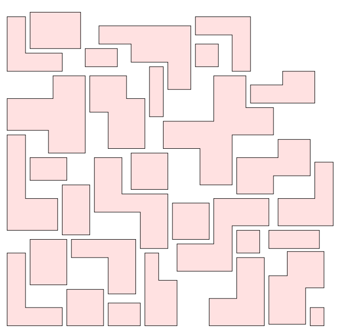
In this paper, we derived an efficient data representation for a special class of rectilinear polygons : ordered convex rectilinear polygons in BSG structure. As such, the x and y dimension can be independently compacted given every polygon is ordered convex shape. By transferring or partitioning arbitrary rectilinear polygons into the ordered convex shapes, the general rectilinear compaction can be dealed with. Furthermore the topology constrained rectilinear block packing is applied to the layout reuse problem. A SP-based BSG assignment is constructed such that the rectilinear blocks can be compacted under the topology constraints.



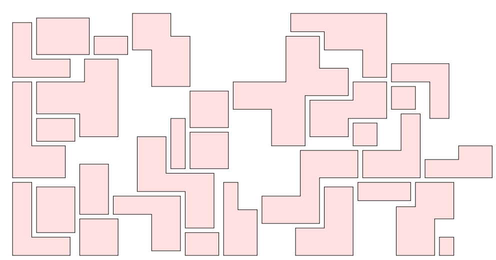
(a) total area = 535 $\times$ 392 = 209,720.



(b) total area = 301 $\times$ 296 = 78,260.



(c) total area = 320 $\times$ 310 = 99,200.



(d) total area = 450 $\times$ 240 = 108,000.

Figure 22: (a) shows the initial placement of 31 rectilinear blocks, each of them has ordered convex shape. (b) shows the packing of 31 blocks achieved by the algorithm presented in this paper, in which the x and y dimension are independently compacted, and the relations of blocks in (a) is preserved. On the other hand, we compact the 31 blocks without considering the topological constraints : (c) shows the packing by first compacting x dimension then y dimension, and (d) shows the packing by first compacting y dimension then x dimension.

Ack
The authors like to thank Shigetoshi Nakatake, H. Murata and K. Fujiyoshi in Tokyo Institute of Technology for helpful discussion. Most of this work was done during the first author's summer internship at Ultima Interconnect Technology Inc.

References

[1] R.H.J.M.Otten, "Automatic floorplan design," in *Proc. of 19th ACM IEEE Design Automat. Conf.*, pp. 261–267, 1982.

[2] D. F. Wong and C. L. Liu, "A new algorithm for floorplan design," in *Proc. 23rd Design Automation Conf.*, pp. 101–107, June 1986.

[3] W.-M. Dai and E. S. Kuh, "Simultaneous floor planning and global routing for hierarchical building-block layout," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, no. 5, pp. 828–837, 1987.

[4] W. W.-M. Dai, B. Eschermann, E. S. Kuh, and M. Pedram, "Hierarchical placement and floorplanning in bear," *IEEE Trans. Computer-Aided Design*, vol. CAD-8, no. 12, pp. 1335–1349, 1989.

[5] T. C. Wang and D. F. Wong, "An optimal algorithm for floorplan area optimization," in *Proc. 27th ACM/IEEE Design Automation Conf.*, pp. 180–186, 1990.

[6] T. C. Wang and D. F. Wong, "A graph theoretic technique to speed up floorplan area optimization," in *Proc. 29th ACM/IEEE Design Automation Conf.*, pp. 62–68, 1992.

[7] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, "Bounded-slicing structure for module placement," Tech. Rep. 313, Institute of Electronics, Information and Communication Engineers of Japan, 1994.

[8] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-packing-based module placement," in *IEEE/ACM International Conf. on Computer Aided Design*, (San Jose, CA), pp. 472–479, November 1995.

[9] M. Kang and W. W.-M. Dai, "Non-slicing floorplanning with l-shaped, t-shaped and soft blocks based on bounded slicing grid," in *Proc. 1997 Aisa and South Pacific Design Automation Conf.*, (Chiba, Japan), pp. 265–270, January 1997.

[10] J. Dufour, R. McBride, P. Zhang, and C.-K. Cheng, "A building block placement tool," in *Proc. 1997 Aisa and South Pacific Design Automation Conf.*, (Chiba, Japan), pp. 271–276, January 1997.

[11] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module Placement on BSG-Structure and IC Layout Applications," in *IEEE/ACM International Cof. on Computer Aided Design*, (San Jose, CA), pp. 484–491, November 1996.