The Multi-BSG: Stochastic Approach to an Optimum Packing of Convex-Rectilinear Blocks

Keishi SAKANUSHI, Shigetoshi NAKATAKE, and Yoji KAJITANI
Department of Electrical and Electronic Engineering
Tokyo Institute of Technology
E-mail:{keishi,nakatake,kajitani}@ss.titech.ac.jp

Abstract

A floorplannar that can handle convex-rectilinear blocks is developed by enhancing the BSG-based packing algorithm. The ideas are in the introduction of (1) multi-rectangle representation of a block as a superpose of element-rectangles, (2) parametric-BSG as a generalization of the BSG, (3) multi-BSG which is an arrangement of plural BSG's on a multi-layer, and (4) layer sharing condition of element-rectangles so that non-overlapping is discussed on each layer. A solution space of packings is defined as the set of packings generated by changing parametric-BSG and room assignments. It is guaranteed to contain an optimal packing if the BSG is not smaller than a certain size. A floorplan based on a simulated annealing was implemented. In experiments, it output highly compressed packings.

1 Introduction

An algorithm to pack the convex-rectilinear blocks is developed as an enhancement of the BSG-based packing algorithm so far proposed in [8, 10]. See Fig.1011 where the packings by the proposing algorithm are shown.

This is aiming to meet the need of the recent advance of microscopic manufacturing technology of semiconductor that yields ultra large scale integrated electronic circuits of the size such that a whole system is capable to be on a single silicon chip. Any competent design methodology in such an environment shall make effective use of the macro-library where macro-cells (blocks) are predesigned in soft- or hard-level such as IP's (intellectual property). Each block is designed for performance (e.g. timing, I/O pins, area, power, and others) so that its shape is more complex than a simple rectangle.

A packing algorithm considering the performance of circuits is called a floorplannar. It determines the locations of blocks in a feedback design style, and a high level synthesis analyzes the layout information to improve the circuit performance. Therefore, the floorplannar is a key tool in layout design as studied in literatures, e.g., [1, 2, 3, 4, 5, 6].

On the other hand, recent development of computation powers enables effective stochastic approaches which output a near optimal solution of a myriad of solutions which have been searched stochastically, as the simulated annealing, genetic algorithm, and others do. The key of a successful stochastic approach is how the solution space is nicely constructed.

In the framework that every input block is a rigid rectangle and the evaluation of the output is $width \times height$ of the bounding box that encloses all the placed blocks, the existing packing algorithms based on the Sequence-Pair or BSG structures are satisfying as reported in [7, 8, 10]. Furthermore, papers [8, 9, 10] demonstrated ideas to pack L- or T-shaped blocks. But their algorithms are ad hoc and very restrictive.

This paper proposes an enhancement of the BSG-based packing algorithm to handle systematically general convex-rectilinear blocks. It is based on several new ideas such as: (1) Multi-rectangle representation of a block in terms of element-rectangles, (2) Parametric-BSG. (3) Multi-BSG, and (4) Layer assignment of element-rectangles.

The set of all the packings generated by the room assignment and junction matrices is the solution space of packings. An important property is that this space is guaranteed to contain an optimal packing.

However, it is impractical to exhaust the space. Hence, a simulated annealing was implemented. Experiments achieved to output highly compressed packings compared with the results by the previous methods[8, 10].

The rest of the paper is arranged as follows. After the preliminaries in Section 2, our proposing packing algorithm is given in Section 3. P-admissibility of the solution space is discussed in Section 4. Experimental considerations are in Section 5. Section 6 is the concluding remarks.

2 Preliminaries

We define the input blocks with their representation and generalize the BSG.

2.1 Multi-Rectangle Representation of Input Blocks

The input is the set $\mathcal{C} = \{c_i\}$ of convex-rectilinear blocks. A convex-rectilinear block is a region enclosed by vertical or horizontal segments such that for any two internal points, a shortest Manhattan path connecting them is included in the region. See Fig.1(left).

Intersection points of horizontal and vertical segments are called apexes, and their locations are represented with respect to the reference point. A special convex-rectilinear block with four apexes is called a rectangle. In this paper, a convex-rectilinear block is called simply a block.

A packing of blocks is a placement of blocks such that no two blocks overlap each other. The area of a packing is defined as the area of the minimum rectangle that bounds all the blocks(bounding box). Then for input $\mathcal{C}$, the packing problem is to get a packing whose area is minimum.

A block is represented by a set $M(c_i) = \{m_{i,j}\}$ of rectangles with locations such that their superpose is exactly the set of inner points of the block. See Fig.1(right). $M(c_i)$ is called the multi-rectangle representation and each rectangle $m_{i,j}$ the element-rectangle of c_i.

The multi-rectangle representation may not be unique for a block. In this paper we use $M(c_i)$ to represent any one if (i) it consists of the minimum number of element-rectangles, and (ii) each element-rectangle is maximal in area.

The number of element-rectangles $|M(c)|$ is referred to as the convexity of c. The convexity is no more than $(number\ of\ convex\text{-}apexes - 3)$, where the convex-apex is an apex whose in-corner angle is $\frac{\pi}{2}$.

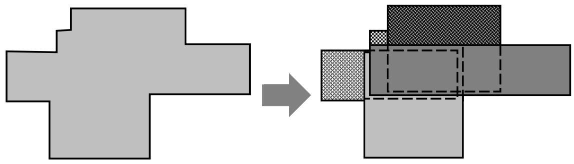


Figure 1: A multi-rectangle representation of a block with 9 convex-apexes by 5 element-rectangles (convexity is 5)

2.2 Parametric-BSG

The Bounded-Sliceline Grid (BSG)[8, 10] consists of vertical and horizontal segments, called BSG-segs, such that the junctions (grid points) are vertically cut or horizontally cut alternatively. $\mathrm{BSG}_{p\times q}$ is consisting of p and q rooms horizontally and vertically respectively. $\mathrm{BSG}_{4\times 4}$ is shown in Fig.2.

Junctions and rooms are given addresses; the most left and top junction is (0,0) and a room has the same address as its left top junction. See Fig.2.

In this paper, we generalize $\mathrm{BSG}_{p\times q}$ by introducing a parameter junction matrix which is

$$T = \begin{pmatrix} t_{0,0} & t_{1,0} & \cdots & t_{p,0} \\ t_{0,1} & t_{1,1} & \cdots & t_{p,1} \\ \vdots & \vdots & \ddots & \vdots \\ t_{0,q} & t_{1,q} & \cdots & t_{p,q} \end{pmatrix}, \quad t_{i,j} = h\ or\ v.$$

This junction matrix T defines $\mathrm{BSG}^T_{p\times q}$, the parametric-BSG, by the following rule.

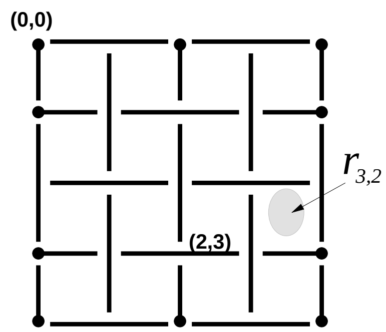
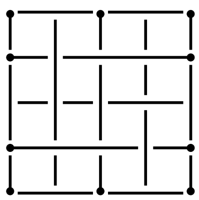


Figure 2: Canonical $\mathrm{BSG}_{4\times 4}$ with addresses of junctions and rooms

Prepare a grid consisting of $p+1$ vertical lines and $q+1$ horizontal lines. The grid point of ith vertical line and jth horizontal line is called the (i,j) junction.

If $t_{i,j} = h$, the vertical line at (i,j) junction is split to make a slit keeping the horizontal line to pass through the slit. If $t_{i,j} = v$, the horizontal line is split similarly. The definition will be clear by an example shown in Fig.3.

Note that the conventional BSG is the one defined by the junction matrix with $t_{i,j}$ being v or h according to $i+j$ even or odd, respectively. We call this the canonical BSG.



$$T = \begin{pmatrix} v & h & v & h & v \\ h & v & h & h & h \\ v & v & v & h & v \\ h & h & h & v & h \\ v & h & v & h & v \end{pmatrix}$$

Figure 3: $\mathrm{BSG}^T_{4\times 4}$, a parametric-BSG

2.3 Multi-BSG

Given k junction matrices $T^1, T^2, \ldots, T^k$, k parametric-BSG's $B^1, B^2, \ldots, B^k$ are defined, respectively. The multi-BSG $\vec{B}^k_{p\times q}$ is a multi-layer structure obtained by superposing all the parametric-BSG's keeping the address of the rooms on each parametric-BSG. It is denoted as

$$\vec{B}^k_{p\times q} = B^1/B^2/\ldots/B^k$$

.

The number k is called the multiplicity. In the following we say that B^l is said to be on the l th layer and r^l is the room on the l th layer. The sequence of rooms with the same address on all the layers is called the multi-room, denoted as $\vec{r}^k_{i,j} = r^1_{i,j}/r^2_{i,j}/\ldots/r^k_{i,j}$. In Fig.4, a multi-BSG is shown.

3 Packing by Multi-BSG

3.1 Outline of Packing

The input is set $\mathcal{C} = \{c_1, ..., c_n\}$ of convex-rectilinear blocks with multi-rectangle representation $M(c_i)$ for each c_i. A packing is obtained by five steps:

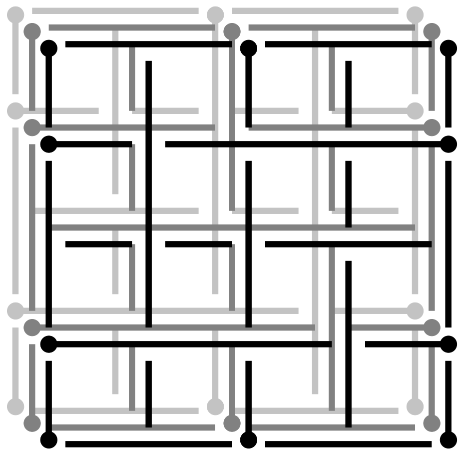


Figure 4: Multi-BSG $\vec{B}^3_{4\times 4}$:Top layered BSG has T in Fig.3

Step 1: Preparation of $\mathbf{BSG}^k_{p\times q}$ with junction matrices T^l $(l = 1, \ldots, k)$.

Step 2: Room assignment by mapping A_R of c_i to distinct rooms.

Step 3: Layer assignment by mapping A_L of element-rectangles to layers of the multi-BSG.

Step 4: Construction of a pair of graphs to represent horizontal and vertical constraints.

Step 5: Compaction .

To execute those steps, we have to fix the following six parameters: $p\times q$, k, T^l, $M(c_i)$, A_R, and A_L.

$p \times q$ must satisfy $p\times q \geq |C|$. Recall that a packing is an arrangement of blocks on the plane under the constraint that any two blocks do not overlap. This is equivalent to the requirement that any two element-rectangles of different blocks do not overlap. This could be verified by using the conventional BSG packing techniques, if a pair of element-rectangles which should not overlap is assigned to distinct rooms on the same layer. This condition on the element-rectangles is formalized as the condition Layer-Share, the discussion on which is postponed to the next subsection. It will be shown that Layer-Share is satisfied if k is not smaller than a certain number, $O(t^2 \log n)$.

Of these six parameters, we fix $p\times q$, k, $M(c_i)$, and A_L. Then the choice of T^l and A_R make the compaction output a unique packing. Our algorithm is a heuristic one to chose a best one by stochastically changing T^l and A_R.

3.2 Room Assignment and Layer Assignment

A room assignment is a one-to-one mapping

$$A_R : \mathcal{C} \rightarrow \overline{R}.$$

where $\overline{R}$ is the set of multi-rooms. Take a block $c_i \in \mathcal{C}$, and assume that it is mapped to a multiroom $\vec{r}^k \in \overline{R}$, i.e. $A_R(c_i) = \vec{r}^k$.

Layer assignment

$$A_L : M(c_i) \rightarrow \vec{r}^k$$

is a one-to-many non-overlapping mapping, that is, a mapping that satisfies

$$A_L(m_{i,\alpha}) \cap A_L(m_{i,\beta}) \quad = \quad \emptyset \text{ if } \alpha \neq \beta$$

In Fig.5, an $M(c_i)$ is shown. Its corresponding A_R and A_L are shown in Fig.6.

The condition Layer-Share is defined for fixed A_R.

Layer-Share: For any pair of distinct blocks c_i and c_j, and for any pair of element-rectangles $m_{i,\alpha} \in M(c_i)$ and $m_{j,\beta} \in M(c_j)$,

$$A_L(m_{i,\alpha}) \cap A_L(m_{j,\beta}) \neq \emptyset$$

i.e. there exists a layer to which $m_{i,\alpha}$ and $m_{j,\beta}$ are assigned.

An example of A_R and A_L is shown in Fig.6 such that Layer-Share is satisfied.

Apparently multiplicity k of the BSG shall be large to satisfy Layer-Share as the convexity and the number n of blocks are large. $k = n(n-1)/2 \times (maxmum\ convexity)^2$ satisfies Layer-Share since we can prepare one layer for each pair of element-rectangles. This is too large for practical implementation. We can prove that a far smaller multiplicity is enough.

Theorem 3.1 *If the maximum convexity is t, there is a layer assignment A_L satisfying Layer-Share if*

$$k = \lceil t(t-1)\log_t n + 1 \rceil.$$

Proof: For the space, it will be shown that we can provide an A_L for the case $t = 2$.

Layer Assignment A_L for $t = 2$

We represent the proposing A_L by a $k \times n$ matrix $\mathbf{A_L}^n$ for $\mathcal{C}$, $|\mathcal{C}| = n$, where the rows correspond to the layers $r^1, r^2, ..., r^k$ in this order and columns the blocks. Let $M(c_i) = \{m_{i,0}, m_{i,1}\}$. The (i,j) element of the matrix is 0 or 1 meaning to assign $m_{i,0}$ or $m_{i,1}$ to the j-th layer.

$\mathbf{A_L}^n$ is defined recursively for $n = 2^x$, $x \geq 2$, starting with $\mathbf{A_L}^4$ as:

$$\mathbf{A_L}^4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad (1)$$

Note that this assignment satisfies Layer-Share since (10), (01), (00),and (11) all appear in any two columns.

Suppose we have got A_L^x, $x \geq 4$ whose size is $k \times x$. Then, define a $(k+2) \times 2x$ matrix.

$$\mathbf{A_L}^{2x} = \begin{pmatrix} \mathbf{A_L}^x & \overline{\mathbf{A_L}}^x \\ \mathbf{1}^x & \mathbf{1}^x \\ \mathbf{0}^x & \mathbf{0}^x \end{pmatrix} \qquad (2)$$

Here, $\overline{\mathbf{A}}$ denotes the matrix obtained from a matrix $\mathbf{A}$ by interchanging 1 and 0, and $\mathbf{1}^x$ and $\mathbf{0}^x$ denote the $1 \times x$ matrices (row vectors) whose elements are all 1 or 0, respectively.

In this construction, the number k of rows is 5 for $n = 4$ and increases by 2 each time n is doubled. Hence $k = 2\log_2 n + 1$ for n in the form $n = 2^x$. For n such as $2^x < n < 2^{x+1}$, we can use any n columns of $\mathbf{A_L}^{2^{x+1}}$. Hence the theorem will be proved if it is shown that $\mathbf{A_L}^n$ satisfies Layer-Share.

It is to prove that for any pair π of columns, there are all the combinations of (00), (11), (10), and (01). $\mathbf{A_L}^4$ shown in eq. (1) trivially satisfies this property as was explained. Assuming that $\mathbf{A_L}^x$ satisfies this property, we prove that A_L^{2x} also satisfies the property. In eq. (2), if both columns of $\pi = (\alpha, \beta)$ are taken from the first x columns (last x columns), the property is satisfied since the submatrix $\mathbf{A_L}^x$ satisfies it. Assume that α is from the first half and β from the last half. (11) and (00) are contained in the added rows $(\mathbf{1}^x\ \mathbf{1}^x)$ and $(\mathbf{0}^x\ \mathbf{0}^x)$. Let β' be the column in the first half corresponding to β. Then, if $\beta' \neq \alpha$, there are (11) and (00) in $\mathbf{A_L}^x$ by assumption. Since the elements of β' are changed in β by construction, there are (10) and (01) at those rows in α and β, completing the proof (when the convexity is 2).

If $\beta' = \alpha$, the proof is similar. □

3.3 Compaction

Let $\vec{B}^k_{p\times q} = B^1/B^2/.../B^k$. Considering each B^l being a single BSG, we can make the horizontal- and vertical-seg constraint graphs. In the multi-BSG, the total horizontal-seg (vertical-seg) constraint graph is constructed by merging those horizontal-seg (vertical-seg) graphs. Finding the longest paths of the resultant graphs, a packing of the convex-rectilinear blocks is obtained. It is described in the following for the horizontal-seg constraint graph only since the vertical-seg one is analogously obtained.

We associate each convex-rectilinear block c_i with a reference point. Each element-rectangle is represented by the 4-tuple as $m_{i,j} = (x_{i,j}, y_{i,j}, w_{i,j}, h_{i,j})$, where the first two denote the relative coordinates with respect to the reference point and the last two width and height.

Horizontal-Seg Constraint Graph $G_h(V_h, E_h)$

1. Construct a horizontal-seg constraint graph $G_h(V^l_h, E^l_h)$ for each layer, which we call the constraint graph of the l *th* layer. Note that each is a directed acyclic graph with a source and a sink.

2. Create a vertex s called the grand source. Replace the sources of all the layer graphs with the grand source. The same change is done for the the sink.

3. Create n vertices $V_c = \{v_i\}(i = 1...n)$, assuming that v_i corresponds to each block c_i.

4. For simplicity, let an arbitrary block c_i be represented as c. Assume that it is assigned by A_R to multi-room $\vec{r}^k$. Also assume that $e^l = (u, v)$ is the edge that crosses the room on the l-th layer where an element-rectangle m_j of c is layer assigned. Execute the following graph change for all $l(= 1, \ldots, k)$. See Fig.7.

   - $E_h \leftarrow (E_h - \{e^l\}) \cup \{(u, v_c), (v_c, u)\}$.
   - Let the weight of edge (u, v_c) be $-y_m$ and that of (v_c, v) be $y_m + h_m$.

□

Find the longest path from the grand source to each vertex in $G_h(V_h, E_h)$. Let the length as the vertical location of the BSG-seg in each layer. Then, the vertical position of the element-rectangle will be determined.

Similarly, the total vertical-seg constraint graph $G_v(V_v, E_v)$ is completed and the horizontal position of the element-rectangle will be determined.

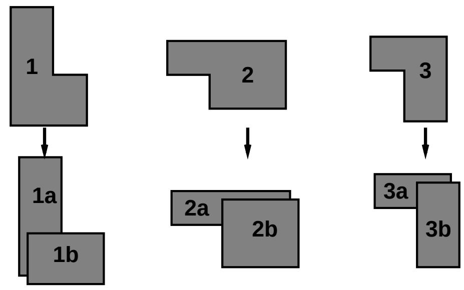


Figure 5: Multi-rectangle representations

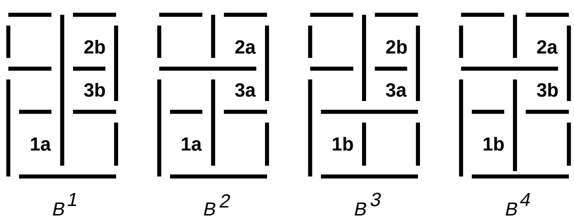


Figure 6: Room and layer assignments

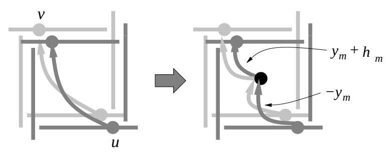


Figure 7: Merging of edges in each layer

In Fig.5 through 8, a total example of element-rectangle decomposition, room assignment, layer assignment up to the graph construction is shown.

The following fact is rather trivial.

Theorem 3.2 *Given a set of blocks, the computational complexity of the compaction procedure is linear to $p \times q$*

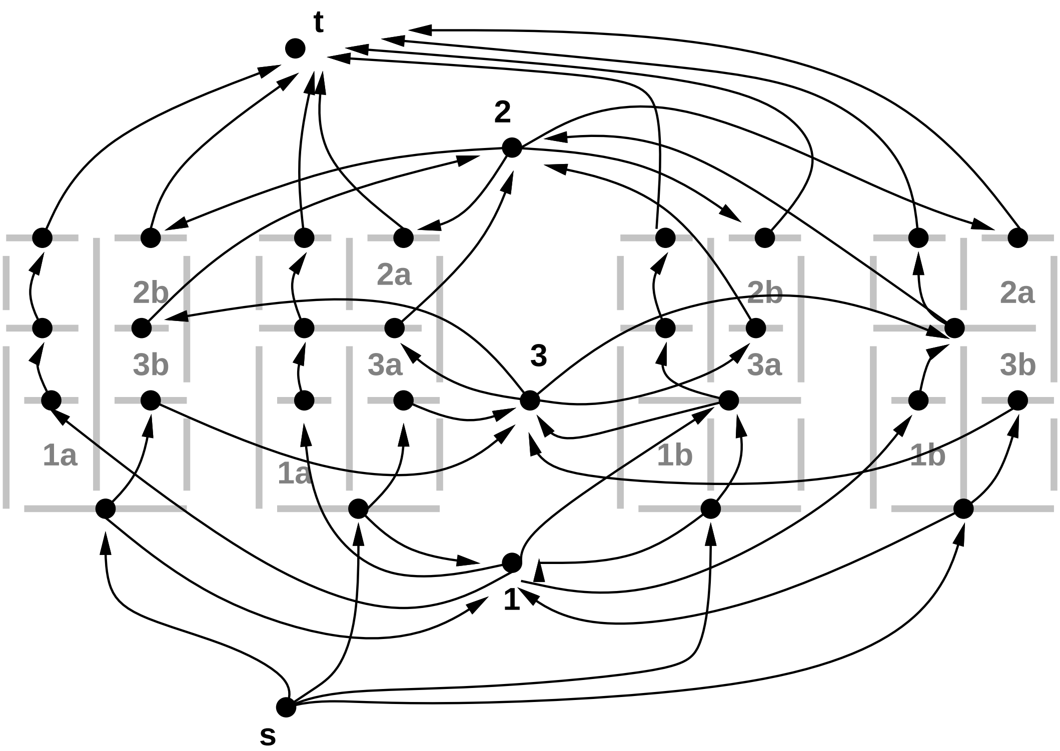


Figure 8: Total horizontal-seg constraint graph

4 P-admissible Solution Space

In packing, with properly defined k and $p \times q$, four varieties were introduced with respect to multi-rectangle representations, junction matrices, room assignments, and layer assignments. Here we fix the multi-rectangle representation and layer assignment of the varieties, and construct a solution space of the packings by changing the room assignment and junction matrix.

The packing algorithm is to search the space to find the best. For the purpose, it is hoped that the solution space is a P-admissible space[7, 8, 10], which satisfies the following four properties;

Solution feasibility: Every solution is feasible, i.e. a packing.

Polynomial-time evaluation: A solution is evaluated in polynomial time.

Space finiteness: The number of solutions is finite.

Optimum containment: An optimal solution is included.

It is obvious from the previous discussion that Solution feasibility and Polynomial-time evaluation are satisfied. The Space finiteness is trivially satisfied, but we would like to know the upper bound.

The number of solutions is the product of varieties of junction matrices and room assignments. It is $2^{p \cdot q \cdot k}$ since the number of junctions of $\vec{B}^k_{p\times q}$ is $p \cdot q \cdot k$ and each junction has a value h or v. The number of distinct room assignments is $\frac{(p\times q)!}{(p\times q-m)!}$. As total, the number of solutions is

$$\frac{(p\times q)!}{(p\times q - m)!} \times 2^{pqk}.$$

For Optimum containment, we do not have any general result on how large k, $p \times q$ are. Even the following simple fact needs a long proof which is omitted here.

Theorem 4.1 *The packing solution space on multi-BSG* $\vec{B}^k_{p\times q}$ *includes an optimal packing if the convexity is 1 or 2 for every block,* $k = \lceil t(t-1)\log_t n + 1\rceil$, *and* $p, q \geq 4 \times |\mathcal{C}|$.

It is believed that the critical values k, p, and q are far smaller than those in the theorem. It is hoped to develop a general algorithm to handle the blocks of arbitrary convexities. However, in practical applications, those values will not be so sensitive to get a satisfactory solution by a stochastic search. In fact, we experienced high compacted packings using the multi-BSG with p, q far smaller than $4|\mathcal{C}|$.

5 Experiments

We restrict the input blocks with convexity being at most 2. We adopted a standard simulated annealing (SA) as a stochastic search.

Letting each BSG be canonical, the initial solution is obtained through a random room assignment. The operation in the search is chosen stochastically from the following four:

1. Interchange of a pair contents of multi-rooms.
2. Rotation of a block.
3. X- or Y-Mirror of a block
4. Change of $t_{i,j}$ of junction matrix. $\{v \to h, h \to v\}$

First, to see the basic performance of our algorithm, we prepared the input data generated artificially, of which we know an optimal packing. It consists of 4 rectangles and 4 L-shaped blocks.

The resultant packing is shown in Fig.9. Multi-rectangle representations, structure of parametric-BSG's room assignment, and layer assignment of the packing, are also indicated.

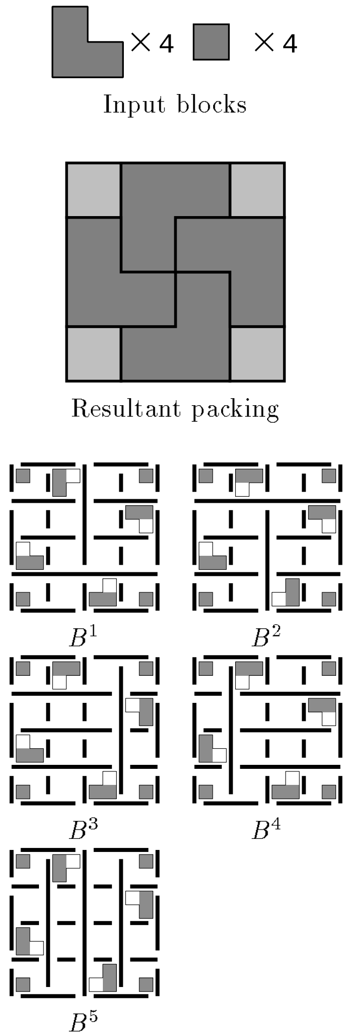


Figure 9: Packing and its state of $\vec{B}^5_{4\times 4}$

Note that the minimal area was attained. Furthermore, it is interesting to observe in the result that the structure of the packing does not contain any sliceline, while each BSG of the multi-BSG is a slicing structure.

Second, we compared our algorithm with the existing one which was proposed for L-shaped block packing in [8, 10]. We used the same data, which consists of 40 blocks including 10 L-shaped blocks.

The area ratio (area of packing divided by sum of area of blocks, denoted by p/r) of the packing in [8, 10] is 112 %, while our algorithm achieved a packing of 109 % area ratio. Our resultant packing is shown in Fig.10. We can see superiority to the existing algorithm[8], which had been then the first proposed algorithm for L-shaped block packing.

Third, we tried to pack more complicated shaped blocks than L- or T-shaped ones. We prepared one data of 10 convex-rectilinear blocks and 40 rectangles which is generated artificially including pathological blocks, and input them to our algorithm. In almost all the cases, we could obtain satisfactory results. One of them is demonstrated in Fig.11.

The CPU time of all the above experiments finished in time of about a quarter day which could be improved considerably by better implementation.

6 Concluding Remarks

It has been hoped a floorplannar that can handle the convex-rectilinear blocks such that designs of

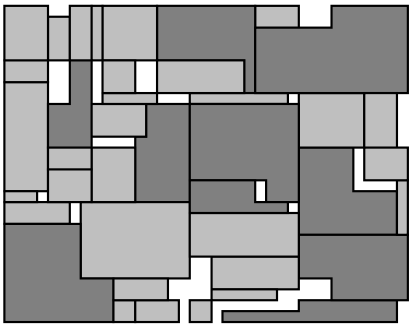


Figure 10: $\vec{B}^6_{20\times 20}$ $(p/r = 109\%)$

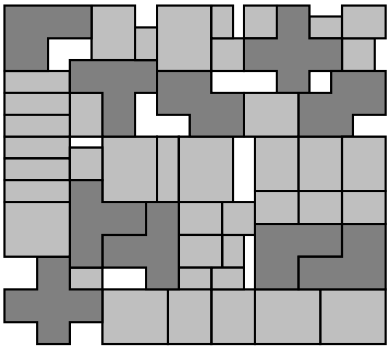


Figure 11: $\vec{B}^6_{15\times 15}$ $(p/r = 111\%)$

macro-cells can be released from the constraint of rectangles and be devoted to the performance of circuits. Our algorithm is described in general framework and its performance is superior to some ad hoc techniques for L- or T- shaped blocks that was proposed in [8, 9, 10].

Future works are to refine Theorems 3.1 and 4.1 including to develop efficient heuristics of layer assignment.

Acknowledgment

Authors are grateful to Dr. Atsushi Takahashi, Associate Professor of Tokyo Institute of Technology, for his helpful suggestion. This research is supported in part by CAD21 and a Grant in Aid for Scientific Research of the Ministry of Education, Science, and Culture of Japan(10750298).

References

[1] R. H. J. M. Otten, "Automatic Floorplan Design," Proc. 19th ACM/IEEE DAC, pp.261-267, 1982.

[2] L.Sha and R.Dutton, "An Analytical Algorithm for Placement of Arbitarily Sized Rectangular Blocks," Proc. of 22nd ACM/IEEE DAC, pp.602-608, 1985.

[3] D. F. Wong and C. L. liu, "A New Algorithm for Floorplan Designs," Proc. 23rd ACM/IEEE DAC, pp.101-107, 1986.

[4] W. Dai, M. Sato, and E. Kuh, "A Dynamic and Efficient Representaion of Building-Block Layout," Proc. of 24th ACM/IEEE DAC, pp.433-439, 1987.

[5] M. Chi, "An Automatic Rectlinear Partitioning Procedure for Standard Cells," Proc. of 24th ACM/IEEE DAC, pp.50-55, 1987.

[6] S. Wimmer, I.Koren, and I. Cederbaum, "Floorplans, Planar Graphs, and Layouts," IEEE Trans. on CAS, pp.267-278, 1988.

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

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

[9] M. Kang and W. Dai, "General Floorplanning with Lshaped, T-shaped and Soft Blocks Based on Bounded slicing Grid Structure," Proc. of ASP-DAC '97, pp.265-270, 1997.

[10] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, "Module Packing Based on the BSG-Structure and IC Layout Applications", to appear in IEEE Trans. on CAD.