IEICE TRANS. FUNDAMENTALS, VOL.E88–A, NO.12 DECEMBER 2005

PAPER Special Section on VLSI Design and CAD Algorithms

A Graph Based Soft Module Handling in Floorplan

Hiroaki ITOGA[†*a)], ***Nonmember***, **Chikaaki KODAMA**[†], ***Student Member***, ***and*** **Kunihiro FUJIYOSHI**[†], ***Member***

SUMMARY In the VLSI layout design, a floorplan is often obtained to define rough arrangement of modules in the early design stage. In the stage, the aspect ratio of each soft module is also determined. The aspect ratio can be changed in the designated range keeping its area of each module. In this paper, in order to determine the aspect ratio, we propose a graph-based one dimensional compaction method which determines the aspect ratio quickly under the constraint that topology of a floorplan must not be changed. The proposed method is divided into two steps: **(1)** Selection of a minimal set of soft modules to adjust the aspect ratio. **(2)** Decision on the aspect ratio. **(1)** is formulated as the minimal cut problem in graph theory. We solve the problem by transforming it to the shortest path problem. **(2)** is divided into two operations. One is to determine the increment limit in height or width of each soft module and the other is to determine the aspect ratio of each soft module by Newton-Raphson method. The experimental comparisons show effectiveness of the proposed method.

key words: *floorplan, soft module, compaction graph, slack*

1. Introduction

In the VLSI layout design, a floorplan is often obtained to define rough arrangement of modules considering relative positions. It is known that a floorplan with nearly optimum area can be obtained within practical time by the search of Simulated Annealing (SA). Generally, modules can be classified into hard modules (with a determined aspect ratio) and soft modules (with permissible multiple aspect ratios). Based on the features of soft modules, the chip area can be reduced effectively. Therefore, in the stage of floorplan optimization, decision of their aspect ratio is desired.

When the relative position of all modules is fixed, it is possible to obtain the optimum aspect ratio of each soft module by nonlinear programming, etc. [1]–[3]. However, the nonlinear programming requires enormous time according to the increase of soft modules since its time complexity is super polynomial order. Therefore, in view of calculation time, it is not practical to apply optimization method using nonlinear programming to each floorplan made in the process of Simulated Annealing.

On the other hand, a method of optimizing the aspect ratio may also be considered after the relative position of all modules is fixed without considering soft modules. However, the optimization of the relative position and of the aspect ratio are carried out separately, so a result might not be expected to be dense. Accordingly, we have no choice but to expand the aspect range impractically or to obtain good results by carrying out this method many times by changing random seeds [3].

One simple method of optimizing the aspect ratio is to implement its random change within a given discrete aspect range as a MOVE operation of Simulated Annealing [4]. This method is easy to implement but it is hard to expect desirable results.

In [5], a heuristic method to optimize the aspect ratio for each floorplan obtained sequentially in Simulated Annealing search was proposed. However, since the selection of soft modules is not appropriate, the chip height may unnecessarily increase when the chip width is reduced. In [6], a complicated method was proposed where soft modules are modified to fit to surplus space in move operation. However, the definition of the method is not clear and the experimental results including soft modules were not shown at all.

In this paper, taking the above into consideration, we propose a quick heuristic method to determine the aspect ratio of each soft module under the constraint of keeping a given relative position between modules. We aim to implement the proposed method with Simulated Annealing. The proposed method is divided into two steps: **(1)** selection of a minimal set of soft modules to adjust the aspect ratio and **(2)** decision on the aspect ratio of elements in the set. **(1)** is formulated as a minimum cut problem in the graph theory, and this is transformed to the shortest path problem to solve efficiently. **(2)** is divided into two problems which are **(2-i)** decision of the increment limit to another direction for each soft module and **(2-ii)** decision of the aspect ratio for each soft module.

The proposed method is applied to each solution (floorplan) generated by MOVE operation on Simulated Annealing search, keeping its relative positions. A computational comparison for searching a good floorplan is carried out.

2. Problem Definition and Organization of the Proposed Method

Floorplanning is a step to plan module arrangements on a chip in the physical design. A **floorplan** (rectangular dissection[**]) is a dissection of a rectangle into a set of rectan-

Manuscript received March 17, 2005.
Manuscript revised June 14, 2005.
Final manuscript received August 1, 2005.

[†]The authors are with the Department of Electric and Electronic Engineering, Tokyo University of Agriculture & Technology, Koganei-shi, 184-8588 Japan.
[*]Presently, with Sony LSI Design Inc.
a) E-mail: itoga@fjlab.ei.tuat.ac.jp

DOI: 10.1093/ietfec/e88–a.12.3390

[**]It is sometimes called "mosaic floorplan" [4].

Copyright © 2005 The Institute of Electronics, Information and Communication Engineers

gles called **rooms** with exclusive assignments of modules to rooms (no two modules share a room). Only T-intersections are used to form the dissection except for the four corners of the outermost bounding rectangle. (Two T-intersections on a point may look like a cross.) A line segment including that of bounding rectangle is called a **seg**.

We consider two kinds of modules: a hard module and a soft module. A **hard module** has specific width and height. A **soft module** has specific area and free width and height to design so far as its aspect ratio is in a given range $[r_{s,\min}, r_{s,\max}]$. In this paper, we consider a problem determining the width and the height of each soft module under the constraint that the relative position of each room pair is equal to a given floorplan.

2.1 Constraint Graph

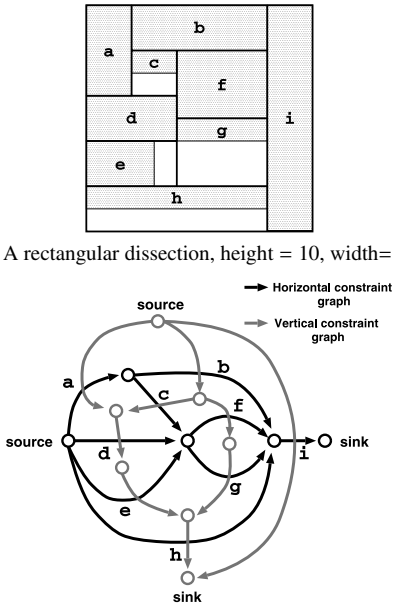
Any floorplan can be modeled using a pair of directed graphs: a horizontal constraint graph and a vertical constraint graph. In the horizontal (vertical) constraint graph, each node represents a vertical (horizontal) seg and each directed edge represents a room. There exists an one-to-one correspondence between the edges and the rooms. A directed edge (u, v) indicates that u is the left (top) side of a room, and v is the right (bottom) side of the room. When the module e is assigned to the room, we call the directed edge "edge e." Each edge has a positive weight representing the width (height) of the module assigned to the room corresponding to each edge. The weight of edge e is represented as the width (height) of module e.

Therefore, both a horizontal and a vertical constraint graphs are directed, acyclic and planar, where parallel edges are allowed, and each graph has a single source and a single sink. The relation between the horizontal and the vertical graphs is polar dual. In a horizontal graph and the corresponding vertical one, each face in one graph corresponds to a node in the other and each edge in one graph crosses a corresponding edge in the other. The longest path length from source to sink in a horizontal (vertical) constraint graph is equal to the minimum width (height) of the chip. An example of a rectangular dissection is shown in Fig. 1(a), and the corresponding constraint graphs are shown in Fig. 1(b). It is easily understood that the following property is derived [7].

Property 1 The number of edges in a horizontal (vertical) constraint graph is equal to the number of rooms. The total nodes in a horizontal constraint graph and a vertical one is equal to the total number of segs and exceeds the number of rooms exactly by three. □

2.2 Organization of the Proposed Method

We will reduce the chip area by adjusting the aspect ratio of soft modules without changing the floorplan. In the proposed method, in order to determine the aspect ratio quickly, a horizontal (vertical) one-dimensional compaction which reduces the chip width (height) is alternately carried out as many times as specified by user but it terminates if the convergence condition is satisfied. The horizontal (vertical) one-dimensional compaction consists of two steps. Firstly, soft modules to adjust their aspect ratio are selected. Then, the aspect ratio of each soft module is decided. In the following, only horizontal one-dimensional compaction is discussed.



(a) A rectangular dissection, height = 10, width= 10.

(b) Corresponding constraint graphs.

Fig. 1 Rectangular dissection and corresponding constraint graphs.

3. Selection of Soft Modules to be Adjusted

In this section, we denote a path consisting of edges $a, b, \cdots, z$ as $[a,\ b,\ \cdots,\ z]$.

3.1 Formulation of the Minimal Cut Problem

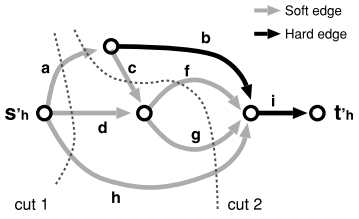
In a horizontal constraint graph, the longest path length from source to sink is equal to the minimum width of a chip, so the chip width gets smaller when the longest path length is reduced. Since a floorplan must be maintained according to the problem definition, only the weight of **soft edges** corresponding to soft modules on the constraint graph can be changed. When there is only a simple longest path, the chip width gets smaller by reducing the weight of soft edges. However, several longest paths may exist, so if the set of soft modules is not selected properly, the longest path length may not be reduced even if the width of soft modules gets smaller, or the width of irrelevant soft modules may be reduced. There is a set of soft modules which can make chip width smaller by reducing the width of the module itself. We call this set of soft modules "*CRM*" (chip reducible soft module set) and it is desirable to obtain the minimal *CRM*.

This set corresponds to **SD-cut** consisting only of soft edges in the sub graph (called **longest path graph** in the following) which is obtained only by leaving edges on the

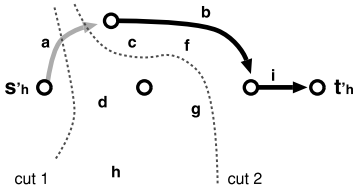
longest path in the horizontal constraint graph and removing other edges. SD-cut in the directed graph is a minimal set of edges which cut off all paths from source to sink [8]. It makes impossible to reach the sink from the source if all edges of SD-cut are removed and makes possible if some and not all edges of SD-cut are removed.

Lemma 1 A minimal *CRM* has one-to-one correspondence to SD-cut consisting only of soft edges on the longest path graph. □

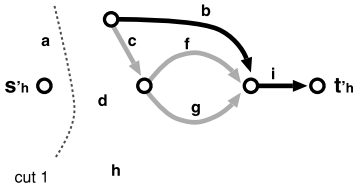
For example, Fig. 2(a) shows a horizontal longest path graph when modules shown in Table 1 are assigned to a rectangular dissection of Fig. 1. In Fig. 2(a), SD-cuts consisting only of soft edges are cut 1 $\{a, d, h\}$ and cut 2 $\{a, f, g, h\}$. If soft edges c, d, f, g and h on the longest path graph are removed (Fig. 2(b)), longest path $[a, b, i]$ remains, and its length does not change. On the other hand, if cut 1 or cut 2 are removed (Figs. 2(c), (d)), all the longest paths are cut off. Note that soft edge c is not included in cut 2 in Fig. 2(d) because its direction is reverse to all edges in cut 2.



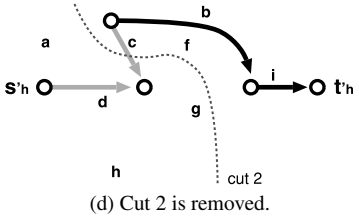
(a) SD-cuts in the longest path graph.



(b) c, d, f, g, h are removed.



(c) Cut 1 is removed.



(d) Cut 2 is removed.

Fig. 2 Horizontal longest path graph corresponding to Fig. 1(a) and the SD-cut consisting of soft edges only.

Table 1 Dimensions of modules for floorplan of Fig. 1(a).

Module	a	b	c	d	e	f	g	h	i
Height	4	2	1	2	2	3	1	1	10
Width	2	6	2	4	3	4	4	8	2
Hard or Soft	S	H	S	S	H	S	S	S	H

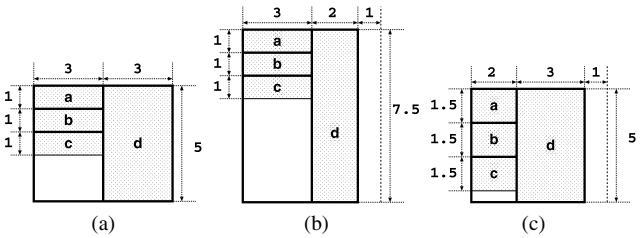
If we want to obtain *CRM* with the minimum number of soft modules, one way is to use **maximum flow and minimum cut algorithm** on network where the capacity of soft edges is one and that of other edges is infinite. However, time complexity is $O(|E||V|^2)$ ($|E|$ is a number of edges, and $|V|$ is a number of nodes) even if the Dinic's algorithm is used. Hence the time complexity is $O(n^3)$ according to Property 1. Here, as for time complexity of other operations of one-dimensional compaction, adjustment of aspect ratios takes $O(n \log n)$ time (mentioned later in Sect. 4), construction of a constraint graph takes $O(n)$ time (mentioned later in Sect. 5), and a search of floorplan takes $O(n)$ time (mentioned later in Sect. 5). Therefore, the maximum flow and minimum cut algorithm would be a bottleneck.

When the chip width is reduced, chip height may increase because the height of some soft modules increase. For example, let's try to reduce the chip width in Fig. 3(a) by one. If width of module d is reduced by one, chip height increases by 2.5 (Fig. 3(b)). On the other hand, if width of modules a, b and c is reduced by one (Fig. 3(c)), chip height does not increase. Therefore, in case of reducing chip width, increments of chip height caused by adjusting the width of soft modules is more important than the number of elements of *CRM*. In addition, increments of the height of soft modules differ even if a reduced width is the same. We want to take this into account, though it may cause an increase of complexity if SD-cut consisting of soft edges is obtained by the maximum flow and minimum cut algorithm.

3.2 Transformation into the Shortest Path Problem

As mentioned above, if we obtain an SD-cut of the longest path graph by the maximum flow and minimum cut algorithm, it takes $O(n^3)$ time, which is a bottleneck.

Hence, based on the method proposed in [9], we obtain SD-cut of the longest path graph as the path of a **compaction graph**, which is defined below. A path on the compaction graph corresponds to an SD-cut consisting only of soft edges. A compaction graph is similar to a polar dual of the longest path graph and can be made from the longest path graph. Then, a set of soft modules to be adjusted is selected by obtaining the shortest path in the compaction



(a) (b) (c)

Fig. 3 Increment of the chip height in reducing the chip width by one. (a) A rectangular dissection (all modules are soft). (b) The width of d is reduced and the chip height increases. (c) The width of a, b and c are reduced but the chip height does not increase.

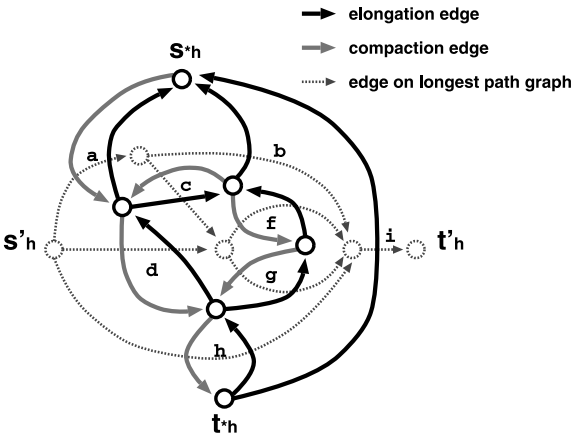


Fig. 4 The vertical compaction graph G_v^* of rectangular dissection shown in Fig. 1(a).

graph.

In the following, we show the algorithm in the horizontal one-dimensional compaction.

Algorithm: Selection of a set of soft modules to be adjusted
Input: A rectangular dissection
Output: Set $\mathcal{M}$ of soft modules to be adjusted
Step 1: Obtain horizontal constraint graph G_h and vertical constraint graph G_v as a planar embedding from a given rectangular dissection (G_h and G_v are planar).
Step 2: Delete edges on G_v crossing those not included in the longest path on G_h. Merge both end points of each deleted edge into a node. We call this resultant graph G_v^-.
Step 3: Turn all edges in G_v^- to the reverse direction. We call them **elongation edges** and set their weights to zero. The edge name of module x is denoted as x'.
Step 4: For each elongation edge in G_v^- whose width can be reduced (namely, it corresponds to a soft module and has a longer width than the minimum decided by the aspect ratio), add an edge of reverse direction to G_v^-. We call the added edges **compaction edges** and set their weights to zero or more. The guideline of "how to set the weights of compaction edges" is: When a soft module has small (large) margins, set large (small) weights to the compaction edge corresponding to it. The resultant graph is a horizontal compaction graph G_v^*.
Step 5: Obtain the shortest path from source to sink in G_v^* using the algorithm of Dijkstra. A set of soft modules corresponding to compaction edges in the obtained path is set $\mathcal{M}$ of soft modules to be adjusted. □

If we reduce as much as possible the width of soft modules having large margins, the chip size will be reduced successfully. Therefore, in **Step 4**, we set the weights of compaction edges freely but set the small weights if the corresponding soft modules have large margins, and set the large weights if they have small margins. It means the edges having small margins are used as much as possible to trace the shortest path.

The example of the compaction graph of Fig. 1(a) is shown in Fig. 4. The paths from the source to the sink in the compaction graph are $[a, d, h]$ and $[a, c', f, g, h]$ that correspond to cut 1 and cut 2 each in Fig. 2.

The fact that a desirable set of soft modules can be obtained by this algorithm depends on the following theorem.
Theorem 1 In the compaction graph G_v^* made from the vertical constraint graph G_v, the following thesis holds. **(1)** On an arbitrary path from source to sink on G_v^*, the set of those soft edges on the longest path graph G_h' which correspond to a set of compaction edges in G_v^* is SD-cut of G_h'. **(2)** On the contrary, for an arbitrary SD-cut consisting of only soft edges of G_h', there is necessarily one or more paths from the source to the sink on G_v^* where the path passes all elements of the set of compaction edges corresponding to the set of soft edges.
Proof **(1)** For all compaction edges and elongation edges on a path (called P) from source to sink on G_v^*, if corresponding edges are cut off on G_h', it is clear that a set of nodes is divided into source side S and sink side D. At this time, in view of nodes at both ends of edges in G_h' to which each edge on P correspond, the node of S is on the right and the node of D on the left facing to the direction of edges on P.

Those edges on G_h' which correspond to elongation edges are necessarily directed from the left node to the right node due to definition. Accordingly, those edges in G_h' which correspond to elongation edges on P are directed from the node of D to the node of S.

Since P consists only of compaction edges and elongation edges, if edges (soft edges) corresponding to compaction edges on P are all removed from G_h', it is impossible to reach the sink from the source.

Since G_h' has a single source and a single sink, there is a path to any node belonging to S from source and a path to the sink from any node belonging to D. Therefore if only one edge belonging to soft edges of G_h' is restored, a path from source to sink in G_h' is necessarily restored. This set of soft edges is minimal and SD-cut.

(2) If G_h' is cut off by an arbitrary SD-cut consisting of only soft edges of G_h', a set of nodes is divided into two, S and D, reachable and unreachable from the source respectively. Clearly S and D are sets of connected nodes since G_h' has a single sink and either of end points of SD-cut does not belong to D.

Let a set of edges whose ends belong to both S and D in G_h' be C. Since S and D are each connected, each of elongation edges in G_v^* which corresponds to each edge of C is a path P from source to sink, if its direction is ignored.

For edge e^* on P corresponding to edge e' from a node of D to that of S, in view of the direction, since e' faces leftward if seen from e^*, the direction of each edge and the corresponding elongation edge agrees. For edge e^* on P corresponding to edge e' from a node of S to that of D, since e' faces rightward if seen from e^*, the direction of each edge and the corresponding elongation edge is reverse to each other. However it is a soft edge, so the direction agrees with

that of compaction edges.

Therefore edges on P are directed paths consisting of all those compaction edges on G_v^* which correspond to the SD-cut and elongation edges. □

Accordingly, the SD-cut consisting only of soft edges in the longest path graph can be obtained by finding the path from source to sink on the compaction graph. The whole time complexity is $O(n \log n)$, because the construction of a compaction graph takes $O(n)$ time and obtaining the shortest path takes $O(n \log n)$ time (it is possible to use Dijkstra's shortest path algorithm, because the weight of edges in the compaction graph was all made to be non-negative). Furthermore, the weight of edges can be set up freely if zero or more, so there is more room for accepting various requests than maximum flow minimum cut algorithm.

In Fig. 1, each soft module in room a, d and h has a small margin of the increment to the vertical direction, and room f and g have a large margin. Hence, according to the guideline mentioned in **Step 4** of the algorithm, the weights of compaction edge corresponding to a, d and h are 1, and f and g are $1/3$[†]. $[a, c', f, g, h]$ is the shortest path, because the length of $[a, d, h]$ is 3, and $[a, c', f, g, h]$ is 8/3. When the method of adjusting aspect ratios proposed in Sect. 4 is applied to soft modules a, f, g and h corresponding to the compaction edge on $[a, c', f, g, h]$, the chip height becomes 10, and the chip width becomes 9.616. On the other hand, if it is applied to soft modules a, d and h corresponding to the compaction edge on $[a, d, h]$, the chip height is 10 and width is 9.674. The selection of $[a, c', f, g, h]$ that is the shortest path makes the chip area smaller.

4. Aspect Ratio of Not Elongating the Chip to the Other Direction

When the chip elongates vertically in the process of horizontal one-dimensional compaction, the chip area may not be reduced even if the horizontal/vertical one-dimensional compaction is alternately repeated. However, it is difficult to obtain the aspect ratio for each soft module that reduces the chip width maximally without elongating the chip height in polynomial time.

We propose a method to determine the aspect ratio for each element of a set of soft modules whose aspect ratios are to be adjusted (called $\mathcal{M}$) by dividing it into two sub problems: The problem of deciding the limit of increment to the other direction for each soft module (detailed in Sect. 4.2) and the problem of deciding the aspect ratio for each soft module on the basis of the given limit of increment.

On the latter, the limit of increment for each soft module is obtained using Newton-Raphson method, because the optimum limit of increment is difficult to obtain in the polynomial time.

In the following, we explain only a horizontal one-dimensional compaction. A vertical one can be done similarly.

4.1 Slack: Limit of Increment to the Other Direction for Each Soft Module

It is well known that *slack* for vertical direction is the maximum amount of increment to the vertical direction without elongation of the chip height, when only one module is increased to the vertical direction. Let s_v be the source, t_v be the sink in the vertical constraint graph, $l(a, b)$ be the longest path length from node a to b, and $w(u, v)$ be the weight of the directed edge (u, v) corresponding to soft module i. Then, slack of i is

$$slack(i) = l(s_v, t_v) - l(s_v, u) - l(v, t_v) - w(u, v). \quad (1)$$

It is clear that the chip height may increase if the height of each element in $\mathcal{M}$ increases by the value of slack of the element.

4.2 Deciding the Aspect Ratio for Each Soft Module

Let the width of soft module i be W_i and the minimal width determined by the limit of aspect ratio be $W_{i,\min}$. Then the range of Δx which is the width reduction of i is $0 \le \Delta x \le W_i - W_{i,\min}$. The vertical increment when width of i is reduced by Δx within this range is

$$f_i(\Delta x) = \frac{W_i \cdot H_i}{W_i - \Delta x} - H_i = \frac{H_i \cdot \Delta x}{W_i - \Delta x} \quad (2)$$

Here H_i is the height of i (area $W_i \cdot H_i$ is fixed).

In the following, we consider the sufficient condition of the vertical elongation value of each module without increasing the chip height. Δy_i represents the upper limit of the vertical elongation value of module i, so $f_i(\Delta x) \le \Delta y_i$.

For all paths from the source to the sink in the vertical constraint graph, if each path length obtained by increasing the height of some modules is less than $l(s_v, t_v)$, the chip height is not increased. Focus on one of the paths (say P). It is easily understood that the chip height does not increase if the following inequality is satisfied.

$$\sum_{j \in Q} \Delta y_j \le \max_{k \in Q} \{slack(k)\} \quad (3)$$

Here Q is the set of modules corresponding to the set of edges on P. Since it is hard to consider the above inequality for the all paths, we consider slack value of all the soft modules and the following inequality for each soft module x is obtained.

$$\sum_{\{j | slack(j) \le slack(x)\}} \Delta y_j \le slack(x) \quad (4)$$

Note that when Eq. (4) is satisfied, Eq. (3) is also satisfied. Therefore, we obtain Δx, reduction of each element i in the set of soft modules $\mathcal{M}$, by solving the following constrained optimization problem.

[†]This is an inverse for *slack* in equality (1).

$$\begin{aligned} &\textbf{Maximize} \quad \Delta x \\ &\textbf{Constraints} \quad 0 \le \Delta x \le \Delta x_{ub} \\ &\qquad \text{for } i \in \mathcal{M} \quad (i = 1, 2, \ldots, |\mathcal{M}|) \\ &\qquad \left\{ \sum_{\{j \mid slack(j) \le slack(i)\}} f_j(\Delta x) \right\} \le slack(i). \end{aligned} \tag{5}$$

If we reduce the width of each element by Δx, the chip width is reduced by at most Δx without elongation of chip height. Here,

$$\Delta x_{ub} = \min_{i \in \mathcal{M}} \{W_i - W_{i,\min}\}$$

should be kept not to break the limit of aspect ratio for each element in $\mathcal{M}$.

When two soft modules have the same slack values, each left side of Eq. (5) is the same and each value of right side of Eq. (5) is the same. Therefore, we divide $\mathcal{M}$ into subsets with the same slack value and name them $Gr_1, Gr_2, \ldots, Gr_k$ in the order of smaller slack values. We denote a slack value of subset Gr as $slack(Gr)$ (obviously $slack(Gr_1) < slack(Gr_2) < \cdots < slack(Gr_k)$). Δx can be obtained by solving the following equations for each of $j = 1, 2, \ldots, k$.

$$\left\{ \sum_{i \in Gr_1 \cup Gr_2 \cup \cdots \cup Gr_j} f_i(\Delta x_j) \right\} = slack(Gr_j) \tag{6}$$

$$\Delta x = \min\{\min_{j=1,2,\ldots,k} \Delta x_j,\ \Delta x_{ub}\} \tag{7}$$

However, Eq. (6) becomes maximally $|\mathcal{M}|$-degree equation which is difficult to solve. Therefore we obtain approximate solution by Newton-Raphson method, iterative solution method of nonlinear equations [10].

5. Experimental Results

We implemented the proposed method in every iteration of annealing process by C++ language. In experiments, we use a Q-sequence [11] to represent a floorplan. Also three move operations are used: (i) ps-move [12] (a move operation proposed for H-sequence, but possible to apply to Q-sequence because of one-to-one correspondence of H-sequence to Q-sequence), (ii) exchange of rooms in the pair, and (iii) hard module rotation. Accepting ratio of each operation is i : ii : iii = 6:3:1. The proposed method is applied to each floorplan generated by the move operations, keeping its relative positions. Time taken for construction of the constraint graphs from Q-sequence is $O(n)$, and for move operation is $O(n)$.

The weight of compaction edge in horizontal one-dimensional compaction is determined as $1/\min\{slack(i), H_{i,max} - H_i\}$ when the height of soft module i corresponding to the compaction edge is H_i and the maximal height determined by the limit of aspect ratio is $H_{i,\max}$. The number of compaction times is set to be five each for horizontal and vertical direction and judgment for convergence is done when no path from the source to the sink exists in both horizontal and vertical compaction graphs.

The experimental comparisons considering only area were carried out for the MCNC benchmarks "apte," "xerox," "hp," "ami33" and "ami49." We used the same range of aspect ratio as in [3]: The aspect ratio for soft modules could be continuously changed within $[1/2,\ 2]$. The results of the proposed method and of [3] are shown in Table 2. Here, "area ratio" represents chip area/total area only of rectangles. The average area is calculated using experimental results obtained from 10 seeds of pseudo-random number. Note that our experiments are performed on a Celeron 468 MHz while [3] used a PentiumIII 600 MHz. The densest results for "ami33" and "ami49" are shown in Figs. 5 and 6 respectively.

In order to confirm the effectiveness for the large scale design, another experiment was carried out using 40 sets of "ami49" (1960 blocks) by Pentium4 2.4 GHz. The aspect ratio could be continuously changed within $[1/2,\ 2]$. The result shown in Fig. 7 is obtained and its area ratio is 105.716%. It is confirmed that the proposed method can obtain good results even if the large scale design is given.

The experimental comparisons considering area and

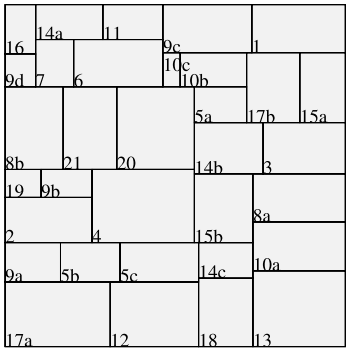


Fig. 5 The best result for "ami33" where all modules are soft, area ratio: 100.003[%], CPU time: 84 [sec].

Table 2 Packing results of the proposed method and the comparison with [3], where all modules are soft (aspect ratio $[1/2,\ 2]$).

Data	#Module	Proposed method (Celeron 468 MHz)		Method in [3] (PentiumIII 600 MHz)	
		Densest (average) area ratio [%]	time [sec]	Area ratio [%]	Time [sec]
apte	9	100.908 (101.061)	23	100.54	53.0
xerox	10	100.218 (100.785)	25	100.4	71.6
hp	11	100.092 (100.602)	26	101.42	107.3
ami33	33	100.003 (100.389)	84	104.49	774.6
ami49	49	100.103 (100.315)	125	108.34	2354.0

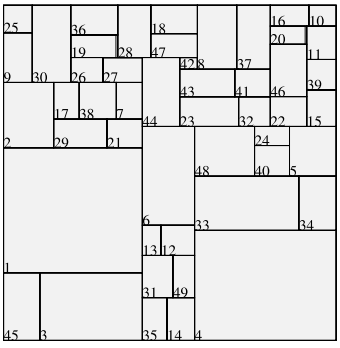


Fig. 6 The best result for "ami49" where all modules are soft, area ratio: 100.103[%], CPU time: 125 [sec].

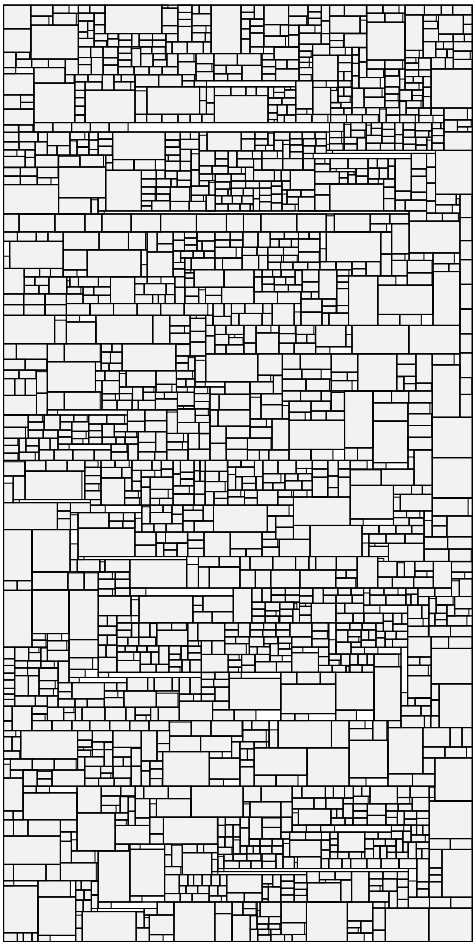


Fig. 7 The best result for ami49 40 sets where all modules are soft (1960 soft modules), area ratio:105.716[%], CPU time: 22779 [sec].

wire length were also carried out for the five benchmarks. We used the same range of aspect ratio as in [1]: The aspect ratio for soft modules could be continuously changed within [1/10, 10]. The evaluating function is the appropriately balanced sum of the resultant area and the total wire length based on the half perimeter approximation for each net, where the terminals are assumed to be at the center of the modules. The best result of the proposed method and of [1] are shown in Table 3. Note that our experiments are performed on a Celeron 468 MHz while [1] used DEC Alpha 250 MHz. In Table 3, all results of the area ratio are inferior to the results in [1] by about one or two percent. However, better results of the wire length except for "apte" were obtained in much shorter time compared with [1]. The best results for "ami33" and "ami49" are shown in Figs. 8 and 9

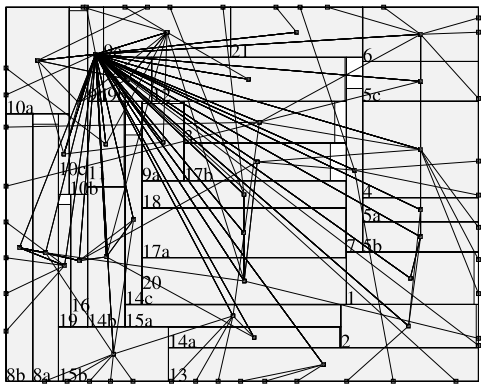


Fig. 8 The best result with wiring for "ami33" where all modules are soft, area ratio: 101.983[%], wire length: 42403, CPU time: 960 [sec].

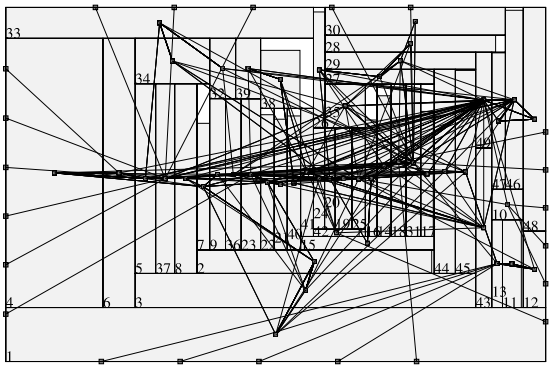


Fig. 9 The best result with wiring for "ami49" where all modules are soft, area ratio: 102.203[%], wire length: 562355, CPU time: 2804 [sec].

Table 3 Packing & wiring results of the proposed method and the comparison with [1], where all modules are soft (aspect ratio [1/10, 10]). (*Including round off error.)

Data	#Module	Proposed method (Celeron 468 MHz)			Method in [1] (DEC Alpha 250 MHz)		
		Area ratio [%]	Wire length	time [sec]	Area ratio [%]	Wire length	Time [sec]
apte	9	100.000	358899	46	99.98*	344358	789
xerox	10	100.940	367381	43	100.82	401254	1198
hp	11	101.035	116300	71	99.96*	118819	1346
ami33	33	101.983	42403	960	100.3	53393	75684
ami49	49	102.203	562355	2804	100.38	775104	612103

respectively.

6. Conclusion

In order to apply each floorplan generated sequentially in Simulated Annealing search process, we proposed a graph-based quick method to determine the aspect ratio of each soft module under the constraint that the relative position of modules (floorplan) is fixed. The proposed method is divided into two steps: selection of soft modules whose aspect ratios are to be adjusted and decision of the aspect ratio for them in the set. The problem of selecting soft modules to be adjusted is formulated as the minimal cut problem in the graph theory, and it is transformed to the shortest path problem. In addition, the problem of the determination of aspect ratios is divided into two sub problems: a problem of the determination of the limit of increment to the other direction and a problem of determination of aspect ratio based on the increment. The experimental comparisons show effectiveness of the proposed method.

References

[1] H. Murata and E.S. Kuh, "Sequence-pair based placement method for hard/soft/preplaced modules," ACM ISPD, pp.167–172, 1998.
[2] K. Fujiyoshi, T. Miwa, H. Murata, and M. Kaneko, "Area minimization for module placement including a novel type of soft-module," IEICE Technical Report, VLD96-104, 1997.
[3] F.Y. Young, C.C.N. Chu, W.S. Luk, and Y.C. Wong, "Handling soft modules in general nonslicing floorplan using Lagrangian relaxation," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.20, no.5, pp.687–692, 2001.
[4] Y. Ma, X. Hong, S. Dong, Y. Cai, C.K. Cheng, and J. Gu, "Floorplanning with abutment constraints based on corner block list," Integration, vol.31, no.1, pp.65–77, 2003.
[5] M. Kang and W.W.M. Dai, "General floorplanning with L-shaped, T-shaped and soft blocks based on bounded slicing grid structure," IEEE ASP-DAC, pp.265–270, 1997.
[6] Y.C. Chang, Y.W. Chang, G.M. Wu, and S.W. Wu, "B*-trees: A new representation for nonslicing floorplans," ACM/IEEE DAC, pp.458–463, 2000.
[7] H. Murata, K. Fujiyoshi, T. Watanabe, and Y. Kajitani, "A mapping from sequence-pair to rectangular dissection," IEEE ASP-DAC, pp.625–633, 1997.
[8] Y. Kajitani, Introduction to the theory of combinatorial algorithms, Corona Publishing, 2002.
[9] S. Takahashi and K. Fujiyoshi, "Compactor for the marking system using sequence-pair," IPSJ SIG Notes, 2002-MPS-38-9, 2002.
[10] J.M. Ortega and W.C. Rheinboldt, Iterative solution of nonlinear equations in several variables, Academic Press, 1970.
[11] K. Sakanushi, Y. Kajitani, and D.P. Mehta, "The quarter-state sequence (Q-sequence) to represent the floorplan," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol.50, no.3, pp.376–386, 2003.
[12] L. Jin, K. Sakanushi, Z. Wu, and Y. Kajitani, "The half-state sequence (H-seq) to represent the channel-adjacency in rooms of a floorplan," IEICE Technical Report, CAS2000-88, 2000.

Hiroaki Itoga received his B.E. degree in electronic and information engineering and M.E. degree in electrical and electronic engineering from Tokyo University of Agriculture & Technology, Tokyo, Japan, in 2000 and 2004, respectively. When he was in university, his research interests were VLSI layout design, especially floor-planning and packing. Currently, he is with Sony LSI Design Inc.

Chikaaki Kodama received his B.E. and M.E. degrees in electronic and information engineering from Tokyo University of Agriculture & Technology, Tokyo, Japan, in 1999 and 2001, respectively. He was with Fujitsu Ltd. from 2001 to 2003. Currently, he is studying toward the D.E. degree in the Department of Electronic and Information Engineering from Tokyo University of Agriculture & Technology. His research interests are VLSI layout design, especially floorplanning and packing, and apparel CAD system. He is a student member of IEEE.

Kunihiro Fujiyoshi received his 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, Ishikawa. He was with Tokyo University of Agriculture & Technology as a lecturer from 1997 and has been an associate professor since 2000 of Department of Electronic and Information Engineering. His research interests are in combinatorial algorithms and VLSI layout design. He is a member of IEEE and IPSJ.