VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair

Hiroshi Murata, *Member, IEEE*, Kunihiro Fujiyoshi, *Member, IEEE*, Shigetoshi Nakatake, *Student Member, IEEE*, and Yoji Kajitani, *Fellow, IEEE*

***Abstract*— The earliest and the most critical stage in VLSI layout design is the placement. The background of which is the *rectangle packing problem*: Given set of rectangular modules of arbitrary sizes, place them without overlap on a plane within a rectangle of minimum area. Since the variety of the packing is uncountably infinite, the key issue for successful optimization is the introduction of a finite solution space which includes an optimal solution. This paper proposes such a solution space where each packing is represented by a pair of module name sequences, called a sequence-pair. Searching this space by simulated annealing, hundreds of modules have been packed efficiently as demonstrated. For applications to VLSI layout, we attack the biggest MCNC benchmark *ami49* with a conventional wiring area estimation method, and obtain a highly promising placement.**

I. Introduction

LAYOUT IN PHYSICAL DESIGN of VLSI is, in short, to pack all the circuit elements in a chip without violating design rules, so that the circuit performs well and the production yield is high. There are so much variety of targets in different stages but the following problem is the core of them.

Rectangle Packing Problem: RP

Let $\mathcal{M}$ be a set of m rectangular modules of fixed orientations, whose heights and widths are given in real numbers. A *packing* of $\mathcal{M}$ is a nonoverlapping placement of the modules. The minimum bounding rectangle of a packing is called the *chip*. Find a packing of $\mathcal{M}$ onto a chip of the minimum area.

A packing of six modules is shown in Fig. 1.

The decision version of our problem **RP(A)** is to decide whether $\mathcal{M}$ can be packed onto a chip of area A. Baker, Coffman, and Rivest [1] proved the $\mathcal{NP}$-completeness of a similar problem **RP(H,W)** : decide whether $\mathcal{M}$ can be packed onto the chip of height H and width W. We can show **RP(A)** to be $\mathcal{NP}$-complete using the fact that any instance of **RP(H,W)** can be polynomially reducible to an instance of **RP(A)** by the following conversion.

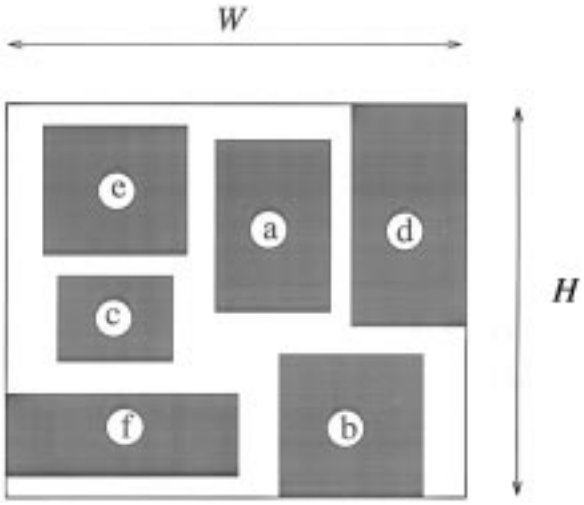

Fig. 1. A packing on a chip $W \times H$.

$$r \leftarrow \frac{\text{the maximum width over modules}}{2H}$$
$$A \leftarrow (W + rH)(W + 2rH)$$
$$\mathcal{M}' \leftarrow \{(w \times rh) | \forall (w \times h) \in \mathcal{M}\} \cup \{rH \times rH, (W + rH) \times (W + rH)\}$$

Our problem **RP** is harder than **RP(A)**, so $\mathcal{NP}$-hard.

Since the heights and widths of modules are real numbers, **RP** is not simply a combinatorial optimization problem. In fact, there have been several numerical approaches [2], [3]. They first generate a possibly overlapping arrangement of modules, and then move modules to reduce the overlapping cost. But the overlap elimination is very hard for the numerical approaches without *ad hoc* post-processing.

An alternative approach is "combinatorial search". In this approach, a set of codes is defined as a *solution space*. Each code represents a construction of placement. A code is said to be *feasible* if the construction is consistent, i.e., there exists a packing corresponding to the code. The evaluation of a feasible code is the area of the chip, and the evaluation of an infeasible code is infinitely negative. The combinatorial search aims at finding a best code in the solution space. However, exhaustive search of the whole space will take too much time. Since the problem is **NP**-hard, the size of any such solution

Manuscript received February 20, 1996; revised September 24, 1996. This work was supported in part by Research Body CAD21 at JAIST. This paper was recommended by Associate Editor C.-K. Cheng.

H. Murata and K. Fujiyoshi are with the School of Information Science, Japan Advanced Institute of Science and Technology, Tatsunokuchi, Ishikawa 923-12 Japan.

S. Nakatake and Y. Kajitani are with the Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Meguro-ku, Tokyo 152, Japan.

Publisher Item Identifier S 0278-0070(96)09413-4.

space is expected to be exponential. Several heuristics have been proposed to find a good solution in a moderate time, for example, simulated annealing and genetic algorithms. Given a time limit, such a heuristic stops the search half-way and outputs the best solution found so far. For this search to be effective, the minimum requirement of the solution space is the following four items.

1) The solution space is finite.
2) Every solution is feasible.
3) Realization of a code is possible in polynomial time.
4) There exists a code which corresponds to one of the optimal solutions.

The solution space that satisfies the above four requirements is called *P-admissible*.

The reasons for 1), 3), and 4) are obvious. That for 2) is most heuristics pick up one solution after another along the neighboring structure defined on the space, consulting with the difference of evaluations (gain) to the previous solution. Therefore, if infeasible solutions are included, the continuity will be destroyed and convergence to a feasible solution is not guaranteed.

A known practical solution space is one derived from the *slicing floorplan* proposed by Otten [4] and others. It satisfies 1), 2), and 3). Several optimization heuristics are applied for the space, and one of the most successful approaches uses simulated annealing [5]. However, since the optimal solution can be nonslicing, 4) is not satisfied. This fact discourages us to start searching for the best in the space. Efforts have been paid to let the space include nonslicing structures [6], [7], but they have not been successful to satisfy 4).

Another approach is proposed by Onodera, Taniguchi, and Tamaru [8]. They construct a solution space by assigning one out of the four relations, "left of," "right of," "above," and "below," to every pair of modules. This space satisfies 4) since any packing satisfies a combination of the relations. But there are many infeasible codes such as; module a is left of module b, b is left of c and c is left of a. Thus their space is not P-admissible either. As a consequence, the space does not admit heuristics such as simulated annealing. In their paper, exhaustive search with a branch-and-bound technique is applied to find an exactly optimal solution, but the size of tractable problems is limited up to six modules.

This paper provides a P-admissible solution space, in which each code is a pair of module name sequences. By searching this space, it has become possible to pack hundreds of modules efficiently, as demonstrated in Figs. 7 and 8.

To utilize this solution space of **RP** for VLSI layout design, the evaluation of a packing has to be modified to consider wires. Some evaluating functions are available for estimating the final chip area [5], [8]. Among them, we use the formula proposed by [8]. The largest MCNC building-block benchmark was successfully placed by simulated annealing in about 30 min (Fig. 9).

This paper is organized as follows. In Section II, a mapping from a given packing to a pair of module name sequences is given. It is proved that at least one of the optimal solutions is included in the space. Section III provides a procedure for an inverse mapping from a sequence pair to a packing. Section IV demonstrates how the space can be utilized in VLSI placement problems. Section V then concludes with final remarks.

An early version of this paper was presented in [9].

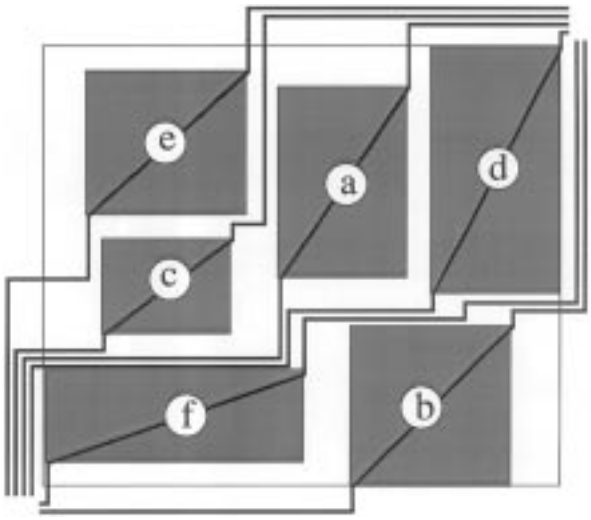

Fig. 2. Positive step-lines.

II. From a Packing to a Sequence-Pair

Let Π be a packing on chip C. See Fig. 1 for an example. We describe a procedure called **Gridding**, which encodes Π to a *sequence-pair*, an ordered pair of module name sequences.

A. Gridding

For each module x, we draw lines using a pebble p (which is regarded as the nib of a pen). The pebble p is initially located at the upper right corner of x and starts to move upward. It turns its direction alternatively right and up until it reaches the upper right corner without crossing: i) boundaries of other modules, ii) previously drawn lines, and iii) the boundary of the chip. The drawn line is called the *up-right step-line* of module x. Similarly, the *down-left step-line* of x is drawn. The union of these two step-lines and the connecting diagonal line of x is called the *positive step-line* of x. It is always possible to draw such positive step-line for a module. They are referred to by the corresponding module names.

An example of resultant positive step-lines is shown in Fig. 2.

From the construction, we have that no two positive step-lines cross each other. It implies that these positive step-lines can be linearly ordered, as well as the corresponding modules. Here we order the positive step-lines from left. Let Γ_+ be the module name sequence in this order.

In Fig. 2, "$\Gamma_+ = ecadfb$" is obtained.

Negative step-lines are drawn similarly as the positive step-lines. The difference is that a negative step-line is the union of the *left-up step-line* and *right-down step-line*, whose direction changing policies are "left, up, left, up, $\cdots$," and "right, down, right, down, $\cdots$," respectively. We order the negative step-lines also from left. Let Γ_- be the module name sequence in this order.

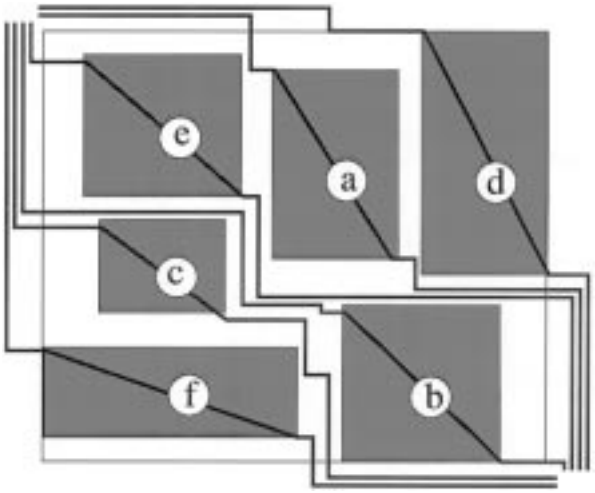

Fig. 3. Negative step-lines.

An example of negative step-lines is shown in Fig. 3. Observing it from left, "$\Gamma_- = fcbead$" is obtained.

The procedure **Gridding** outputs the sequence-pair (Γ_+, Γ_-).

B. Information of the Sequence-Pair

Let (Γ_+, Γ_-) be the sequence-pair produced by **Gridding** for a packing Π. Modules x and x' are related in exactly one of four ways: x' is after/before x in Γ_+/Γ_-. Let us define four disjoint subsets of $\mathcal{M}$, accordingly.

$\mathcal{M}^{\mathrm{aa}}(x) = \{x' \mid x' \text{ is after } x \text{ in both } \Gamma_+ \text{ and } \Gamma_-\}$,
$\mathcal{M}^{\mathrm{bb}}(x) = \{x' \mid x' \text{ is before } x \text{ in both } \Gamma_+ \text{ and } \Gamma_-\}$,
$\mathcal{M}^{\mathrm{ba}}(x) = \{x' \mid x' \text{ is before } x \text{ in } \Gamma_+ \text{ and after } x \text{ in } \Gamma_-\}$,
$\mathcal{M}^{\mathrm{ab}}(x) = \{x' \mid x' \text{ is after } x \text{ in } \Gamma_+ \text{ and before } x \text{ in } \Gamma_-\}$.

For example, with respect to the sequence-pair $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$, four subsets for module c are $\mathcal{M}^{\mathrm{aa}}(c) = \{a, b, d\}$, $\mathcal{M}^{\mathrm{bb}}(c) = \emptyset$, $\mathcal{M}^{\mathrm{ba}}(c) = \{e\}$, and $\mathcal{M}^{\mathrm{ab}}(c) = \{f\}$.

Any module other than x belongs to a unique subset, and it is trivial that two modules are in a dual relation through $x \leftrightarrow x'$, and $a \leftrightarrow b$ as

$$x' \in \mathcal{M}^{\mathrm{aa}}(x) \Leftrightarrow x \in \mathcal{M}^{\mathrm{bb}}(x')$$
$$x' \in \mathcal{M}^{\mathrm{ba}}(x) \Leftrightarrow x \in \mathcal{M}^{\mathrm{ab}}(x')$$

In a packing, if the right side of module x is left of the left side of module x', x is said to be *left of* x'. Similarly, *right of*, *above*, *below* relations between two modules are defined.

Theorem 1: Let (Γ_+, Γ_-) be the sequence-pair produced by **Gridding** for a packing Π. If $x \in \mathcal{M}^{\mathrm{bb}}(x')$, then x is left of x' in Π. The claim also holds when the pair of words ("$\mathcal{M}^{\mathrm{bb}}$" and "left of") is replaced by any of ("$\mathcal{M}^{\mathrm{aa}}$" and "right of"), ("$\mathcal{M}^{\mathrm{ba}}$" and "above"), and ("$\mathcal{M}^{\mathrm{ab}}$" and "below").

In the previous example, module b is in $\mathcal{M}^{\mathrm{aa}}(c)$. One can examine b is actually right of c in the packing shown in Fig. 1.

Proof: Let x and x' be arbitrary two modules. The step-lines of x divide the chip into four regions. Among them, the region surrounded by the up-right step-line of x and the right-down step-line of x together with the right side of the chip is called the *right-cone* of x. Analogously, the *left-*, *above-*, and *below-cone* denote the other three regions.

Suppose x' is in $\mathcal{M}^{\mathrm{aa}}(x)$. This implies that the positive step-line of x' is in the union of the right-cone and the below-cone of x. Also it is implied that the negative step-line of x' is in the union of the right-cone and the above-cone of x. The cross point of the positive and negative step-lines of x' is in their intersection, that is, the right-cone of x. Then, module x' is in the right-cone of x. Every modules in the right-cone of x is right of module x by definition of the up-right step-line and the right-down step-line of x.

It is clear that the claim holds for the other cases. □

III. From a Sequence-Pair to a Packing

In the previous section, we analyzed the packing and fixed the procedure **Gridding** to obtain one sequence-pair from a given packing. Now we provide a procedure to synthesize one packing from an arbitrary sequence-pair.

A. Constraint of a Sequence-Pair

Given a sequence-pair (Γ_+, Γ_-), we read a constraint from it as follows.

The Constraint Implied by a Sequence-Pair (Γ_+, Γ_-)

If $x \in \mathcal{M}^{\mathrm{bb}}(x')$, module x must be left of module x'. This is also the constraint with replacing the pair of words ("$\mathcal{M}^{\mathrm{bb}}$" and "left of") with any of ("$\mathcal{M}^{\mathrm{aa}}$" and "right of"), ("$\mathcal{M}^{\mathrm{ba}}$" and "above"), and ("$\mathcal{M}^{\mathrm{ab}}$" and "below").

It is easily seen that the constraint imposed on the packing by a sequence-pair is unique. Furthermore, the following theorem holds.

Theorem 2: The constraint is always satisfiable.

Proof: Consider an $m \times m$ grid. Label the horizontal grid lines and vertical grid lines with module names along Γ_+ and Γ_- from top and from left, respectively. A cross point of the horizontal grid line of label x and the vertical grid line of label x' is referred to by (x, x'). Then, rotate the resultant grid by 45 degrees counter clockwise to get an oblique grid. (See Fig. 4.) Put each module x with its center being on (x, x). Expand the separation of grid lines enough to eliminate overlapping of modules. (The expansion is enough if the separation is $\sqrt{2}$ times larger than the longest width/height over modules.) The resultant packing trivially satisfies the constraint implied by the given sequence-pair. □

An example is shown in Fig. 4.

B. The Best Packing under the Constraint

Given (Γ_+, Γ_-), one of the optimal packings under the constraint can be obtained in $O(m^2)$ time by applying the well-known *longest path algorithm* for vertex weighted directed acyclic graphs. The process is given below.

Based on "left of" constraint of (Γ_+, Γ_-), a directed and vertex-weighted graph $G_H(V, E)$ (V: vertex set, E: edge set), called the *horizontal-constraint graph*, is constructed as follows.

1) V: source s, sink t, and m vertices labeled with module names

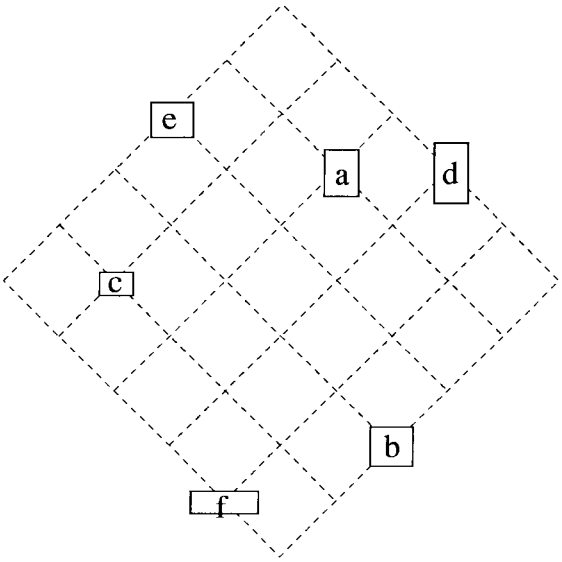

Fig. 4. A packing on an oblique grid for $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$.

2) E: (s, x) and (x, t) for each module x, and (x, x') if and only if $x \in \mathcal{M}^{\mathrm{bb}}(x')$ ("left of" constraint)
3) Vertex-weight: zero for s and t, width of module x for the other vertices

Similarly, the *vertical-constraint graph* $G_V(V, E)$ is constructed using "below" constraint and the height of each module.

Neither of these graphs contains any directed cycle. We set the X-coordinate of x to be the longest path length from s to x in G_H. The Y-coordinate of x is set independently using G_V. If two modules x and x' are in horizontal relation, then there is an edge between x and x' in G_H, hence they do not overlap horizontally in the resultant placement. Similarly, if x and x' are in vertical relation, they do not overlap vertically. Thus no two modules overlap each other in the resultant placement because any pair of modules are either in horizontal or vertical relation.

The width and the height of the chip is determined by the longest path length between the source and the sink in G_H and G_V, respectively. Since the width and the height of the chip is independently minimum, the resultant packing is the best of all the packings under the constraint. The longest path length calculation on each graph can be done in $O(m^2)$ time, proportional to the number of edges in the graph.

As an example, G_H and G_V are shown in Fig. 5 for $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$. The resultant placement after the longest path length calculation is shown in Fig. 6.

C. The P-admissible Solution Space

Previous discussions conclude:

Theorem 3: The set of all sequence-pairs is a P-admissible solution space of **RP**. More precisely, it consists of $(m!)^2$ sequence-pairs, each of which can be mapped to a packing in $O(m^2)$ time, and at least one of which corresponds to one of the optimal solutions of **RP**. □

Our discussion started for minimizing the area of the chip. However, all the discussions hold as long as the evaluating function is independently nondecreasing with respect to the width and the height of the chip. Therefore, we may assume instead, for example, perimeter of the chip, area of the chip of prespecified aspect ratio, and the height of the chip when its width is fixed. This fact will extend the usefulness of our solution space.

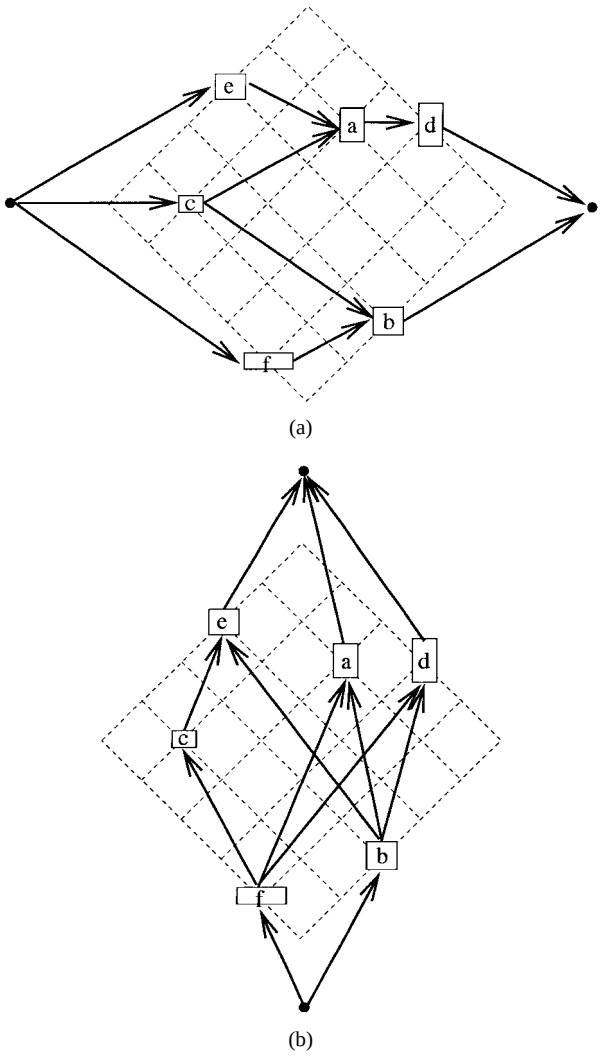

Fig. 5. Constraint graphs G_H (above) and G_V (below) (transitive edges are not drawn for simplicity).

It has also been assumed that the orientation of each module (vertically laid or horizontally laid) is fixed. When the orientation is also requested to be optimized, we hold a {0,1} sequence of length m, expressing the orientation of each module being horizontal or vertical. The size of the solution space increases to $(m!)^2 2^m$. (The orientation optimization for a fixed floorplan is known to be NP-hard [10].) This technique can be easily extended to so-called "soft" modules, by preparing three or more candidates of (width, height) pairs per module [11].

There is a sequence-pair for which another sequence-pair provides no worse packing, independent of the sizes of the modules. For example, if $(abcd, cdab)$ corresponds to an optimal packing, then $(abcd, cadb)$ or $(acbd, cdab)$ also corresponds to an optimal packing, regardless of the widths and the

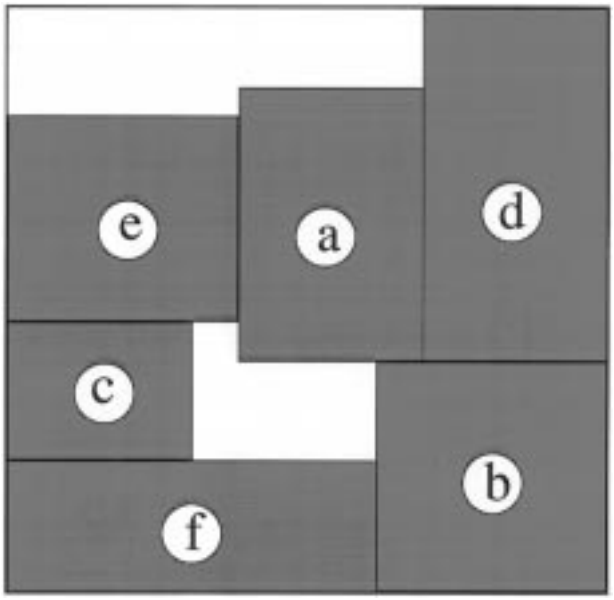

Fig. 6. A best packing under the constraint implied by $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$.

heights of modules a, b, c and d. Then, the former sequence-pair, $(abcd, cdab)$, is redundant for our current objective to find a packing with smaller area. We extend our evaluating function to consider wires in the next section.

IV. Use of the Solution Space and Experiments

A. Rectangle Packing

To show the usefulness of the proposed solution space, dimensions of 146 modules were extracted from a printed circuit board in a personal computer, and packed by standard simulated annealing method. It uses three kinds of pair-interchanges: i) two module names in Γ_+, ii) two module names both in Γ_+ and Γ_-, and iii) the width and the height of a module, where the last one is for orientation optimization. The initial sequence-pair was made as $\Gamma_+ = \Gamma_-$, which corresponds to a linear horizontal arrangement of modules. The temperature was decreased exponentially. From a heuristic point of view, operation i) was selected with higher probability in higher temperature, and operation iii) was selected with higher probability in lower temperature.

The result is shown in Fig. 7. Computation on Sun Sparc-StationII stopped in 29.9 min reaching the terminating temperature. The algorithm searched at most 606 192 distinct sequence-pairs out of the solution space of size $(146!)^2 2^{146} \sim 1.23 \times 10^{552}$. Notice that, the search of only a fraction about 4.92×10^{-547} of the solution space was enough to obtain the placement shown in Fig. 7.

As another challenge, we tried 500 modules, using 18.83 h to get the result shown in Fig. 8.

B. Module Placement with Wires

For VLSI placement, we extend the evaluation to consider wires. Among various possible evaluations about wires, we focus on the final chip area after all wires are detailed-routed.

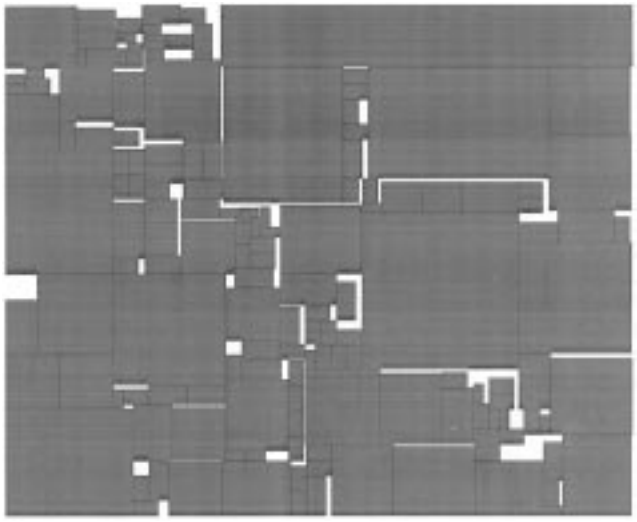

Fig. 7. Packing of 146 modules.

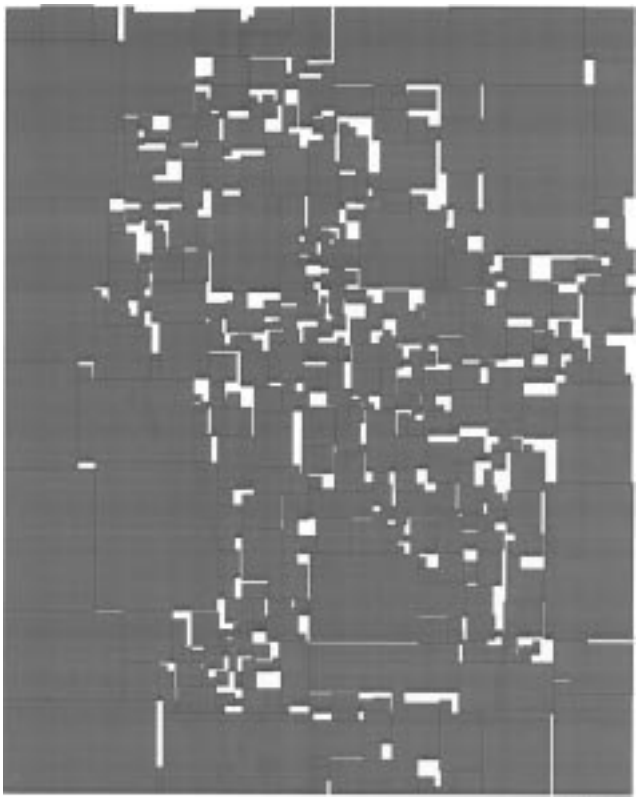

Fig. 8. Packing of 500 modules.

However, it is difficult at this moment and hence we made use of an estimate of the final chip area without the actual routing phase.

Let (Γ_+, Γ_-) be a sequence-pair, Π be one of the optimal packings under the constraint implied by (Γ_+, Γ_-), and W and H be the width and the height of Π. Terminals are given as fixed points on the boundary of each module. A *net* is a set of terminals (multiterminal net), which must be connected by wires, later in detailed-routing phase. A set of nets is given as the netlist $\mathcal{N}$. Given terminals of a net i, which spread over Π, the width and the height of the smallest bounding box of

these terminals are denoted by W_i and H_i, respectively. T is the sum of wiring width and wiring space (obtained from a design rule set). We use the following formula proposed in [8] to estimate the final chip width W' and height H'.

$$W' = W + T\frac{\Sigma_{i\in\mathcal{N}}H_i}{H}$$
$$H' = H + T\frac{\Sigma_{i\in\mathcal{N}}W_i}{W}$$

The second term of each formula estimates the increase in one direction owing to the wires, assuming all wires are uniformly distributed in the final chip. They experimentally showed that the result is acceptable for a commercial channel router [8].

There are a total of eight choices per module, which is the combination of the four choices of 0°, 90°, 180°, 270° rotations, and a decision {*yes, no*} on reflecting the module about the Y axis. This code for orientation and a sequence-pair are put together into a simulated annealing process in our system. The process runs in a similar fashion as the rectangle packing optimization, and explores the solution space of size $(m!)^2 8^m$.

A point not mentioned in [8] is how the location of each individual module is calculated. In our system, after the best evaluated code is obtained, coordinates of each module are determined as follows. Assume (X_j, Y_j) is the coordinates of the lower left corner of module j in Π. (This is the information we can use in this phase.) Let $\mathcal{N}_{X_j}$ be a set of nets such that the X coordinate of the left side of bounding box of the net is less than or equal to X_j. Similarly, $\mathcal{N}_{Y_j}$ is defined using Y_j. We determine the coordinates (X'_j, Y'_j) of the lower left corner of module j in the resultant chip by the following formula:

$$X'_j = X_j + T\frac{\Sigma_{i\in\mathcal{N}_{X_j}}H_i}{H}$$
$$Y'_j = Y_j + T\frac{\Sigma_{i\in\mathcal{N}_{Y_j}}W_j}{W}.$$

For an experiment, the biggest building block layout data, called "ami49," was taken from the MCNC benchmarks. The data is the biggest one in their benchmark suit, but our method was fast enough to handle the data without splitting the problem. The result is shown in Fig. 9. We remark that it was done with an additional constraint: aspect ratio $= 1$, taking $T = 7$ μm. The estimated chip size is 6482 μm $\times$ 6925 μm. Computation time was 31.36 minutes on SunIPX. (A recent study [12] also handled the data without dividing the problem.)

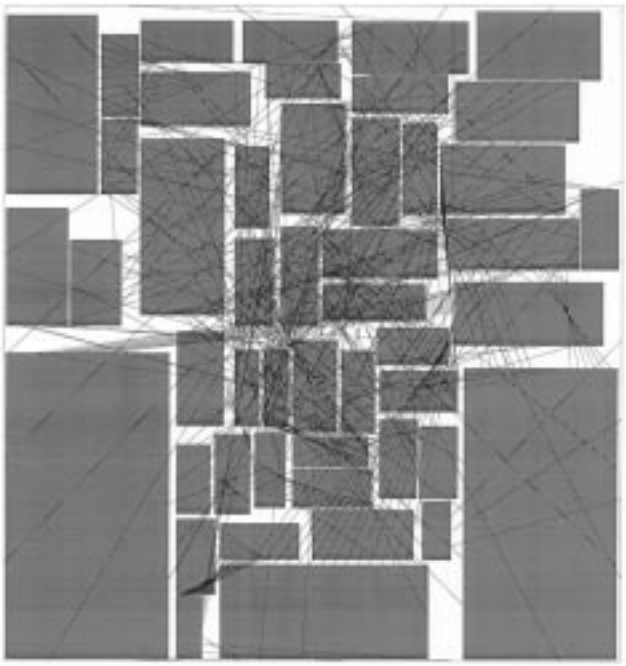

Fig. 9. Placement of MCNC "ami49".

V. Concluding Remarks

The motivation for this work was our experience that many VLSI designers are not satisfied with the slicing structure. This paper introduced a representation of a general packing in terms of a pair of module name sequences, called sequence-pair. Detailed proofs are presented to show that every sequence-pair feasibly corresponds to a packing, and at least one sequence-pair corresponds to an area-minimum packing.

In experiments, 500 rectangles were packed very efficiently in a reasonable time. It was attained by standard simulated annealing in which a move is a change of the sequence-pair. The evaluating function was then extended to the VLSI placement problem using a conventional wiring area estimation method proposed by [8]. The biggest MCNC benchmark, ami49, is placed promisingly.

New VLSI technologies are offering a sufficient number of layers for routing to allow enough space for routing to be done on top of the modules, assuming the use of some "area router". This technology trend would bring placement design still closer to packing, hence make our approach more significant.

Practical layout in VLSI, PCB, or analog circuit design is processed under various constraints and different objectives. For example, wire congestion estimation, timing, power dissipation, clock distribution, crosstalk, are crucial to consider. To include them in our pure packing algorithm, many studies on real instances and theories are necessary.

Another our concern is to make the advantages and disadvantages of the packing based approach more concrete compared with the slicing structure based approach. Though generality is limited, the slicing structure has been known to have a number of advantages, including safe routing order [13], and channel definition. It is left for future to include such properties in the packing based approach.

Acknowledgment

The authors would like to thank Prof. M. Halldórsson for numerous helpful comments.

References

[1] B. S. Baker, E. G. Coffman, and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM J. Comput.*, vol. 9, no. 4, pp. 846–855, 1980.

[2] L. Sha and R. W. Dutton, "An analytical algorithm for placement of arbitrarily sized rectangular blocks," in *Proc. 22th ACM/IEEE Design Automation Conf.*, 1985, pp. 602–608.
[3] A. Alon and U. Ascher, "Model and solution strategy for placement of rectangular blocks in the euclidean plane," *IEEE Trans. Computer-Aided Design*, vol. 7, pp. 378–386, 1988.
[4] R. H. J. M. Otten, "Automatic floorplan design," in *Proc. 19th ACM/IEEE Design Automation Conf.*, 1982, pp. 261–267.
[5] D. F. Wong and C. L. Liu, "A new algorithm for floorplan designs," in *Proc. 23rd ACM/IEEE Design Automation Conf.*, 1986, pp. 101–107.
[6] W. M. Dai and E. Kuh, "Simultaneous floorplanning and global routing for hierarchical building block layout," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 828–837, 1987.
[7] T. C. Wang and D. F. Wong, "An optimal algorithm for floorplan area optimization," in *Proc. 27th ACM/IEEE Design Automation Conf.*, 1990, pp. 180–186.
[8] H. Onodera, Y. Taniguchi, and K. Tamaru, "Branch-and-bound placement for building block layout," in *Proc. 28th ACM/IEEE Design Automation Conf.*, 1991, pp. 433–439.
[9] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-packing-based module placement," in *IEEE Int. Conf. Computer-Aided Design*, 1995, pp. 472–479.
[10] L. Stockmeyer, "Optimal orientations of cells in slicing floorplan designs," *Inform. Contr.*, vol. 59, pp. 91–101, 1983.
[11] P. Pan, W. Shi, and C. L. Liu, "Area minimization for hierarchical floorplans," in *IEEE Int. Conf. Computer-Aided Design*, 1994, pp. 436–440.
[12] H. Esbensen and E. S. Kuh, "An MCM/IC timing-driven placement algorithm featuring explicit design space exploration," in *Proc. IEEE Multi-Chip Module Conf.*, 1996, pp. 170–175.
[13] Y. Kajitani, "Order of channels for safe routing and optimal compaction of routing area," *IEEE Trans. Computer-Aided Design*, vol. 2, pp. 293–300, 1983.

Hiroshi Murata (M'95) was born in 1957. He received the B.E. degree in electrical engineering from Kanazawa University, Ishikawa, Japan, in 1980, and the M.E. degree in information science from Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, in 1994. He is currently working toward the Ph.D. degree at JAIST.

He joined Computer Applications Co., Ltd., Tokyo, Japan, in 1980, and moved to Komatsu Murata Mfg. Co., Ltd., Ishikawa, Japan, in 1984. He has been working on customizing their CAD environment for their Hybrid IC and PCB products. His research interests are in CAD algorithms for PCB's and VLSI's.

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

He has been a research associate with the School of Information Science at the Japan Advanced Institute of Science and Technology since 1992. His research interests are in combinatorial algorithms and VLSI layout design.

Shigetoshi Nakatake (S'94) was born in 1969. He received the B.E. degree in electrical and electronic engineering from Tokyo Institute of Technology, in 1992 and M.E. degree in information science from the Japan Advanced Institute of Science and Technology in 1994. He was at the doctoral program of the Institute until 1996.

Since 1996, he has been working as a research associate with the Department of Electrical and Electronic Engineering, Tokyo Institute of Technology. His current research interests are in VLSI layout design and combinatorial algorithms.

Yoji Kajitani (F'92) was born in 1941. He received the B.S., M.S., and D.E. degrees from the Tokyo Institute of Technology, all in electrical engineering, in 1964, 1966, and 1969, respectively.

He has been a Professor with the Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, since 1985. He was a Professor with the Japan Advanced Institute of Science and Technology from 1992 to 1995. His current research interest is in combinatorial algorithms applied to VLSI layout design.