A New Encoding Scheme for Rectangle Packing Problem

Toshihiko Takahashi

Graduate School of Science and Technology
Niigata University
Niigata, Niigata 950-2181, JAPAN
Tel: 025-262-6721
Fax: 025-262-6721
e-mail: takahasi@ie.niigata-u.ac.jp

Abstract— In the rectangle packing problem, encoding schemes to represent the placements of rectangles are the key factors of efficient algorithms. In 1995, an epoch-making encoding scheme, known as SEQ-PAIR, was developed[2]. The solution space of SEQ-PAIR has been considered sufficiently small. In this paper, however, we present a simple and natural encoding scheme of rectangle packings whose solution space is smaller than that of SEQ-PAIR.

I. Introduction

In the physical design of VLSI, we often have to place a set of modules on a small chip. This naturally leads us to the following problem.

Rectangle Packing Problem (RP) : For given set of n rectangles $\{r_1, r_2, \ldots, r_n\}$, find a packing of minimum area under the condition that the orientation of each rectangle is fixed.

A *packing* is a non-overlapping placement of all rectangles in S. In our RP, each rectangle r_i has width w_i and height h_i in real numbers and the area of a packing is defined as the area of its bounding rectangle. Fig.1 shows an example of a packing. (The bounding rectangle encloses all the shaded rectangles.) RP is known to be NP-hard[1, 2].

A *code* is a representation of a packing under some encoding scheme. The *solution space* of an encoding scheme is defined as the set of all codes.

For example, consider a straightforward encoding scheme which represents a packing as a sequence of the south-west corners (x_i, y_i) of the rectangles r_i in the placement. Obviously, the solution space of this encoding scheme is an infinite set. Note that every code does not have the corresponding packing.

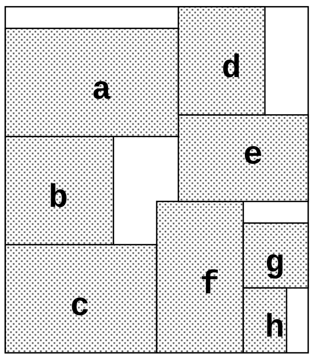

Fig. 1. A packing and its bounding rectangle

In 1995, Murata et al. developed an epoch-making encoding scheme of packings for combinatorial search approach, that is SEQ-PAIR [2]. (SEQ-PAIR is pronounced as "sequence-pair"). For a given set of rectangles $S = \{r_1, r_2, \ldots, r_n\}$, an ordered pair of rectangle sequences represents a packing under SEQ-PAIR.

Therefore, SEQ-PAIR transforms RP into a combinatorial search problem. The size of the solution space of SEQ-PAIR is exactly $(n!)^2$ and a packing algorithm searches this finite solution space for an optimal solution.

SEQ-PAIR has reduced the size of solution space from infinite to finite. In this paper, however, we present a simple and natural encoding scheme LOT, under which a packing is represented by a labeled ordered tree. LOT also has good properties as SEQ-PAIR and its solution space is smaller than that of SEQ-PAIR.

II. SEQ-PAIR

A packing is represented as an ordered pair of permutations of n rectangle names under SEQ-PAIR. For example, $(\Gamma_+, \Gamma_-) = (abcdefgh, cbfhgaed)$ is a sequence-pair corresponding to the packing in Fig.1. We can encode any packing into a sequence-pair[2].

A sequence-pair has information on horizontal and vertical constraint for every pair of rectangles, that is,

(1) r_i is placed left of r_j if r_i is left of r_j in both Γ_+ and Γ_-,
(2) r_i is placed above r_j if r_i is left of r_j in Γ_+ and right of r_j in Γ_-.

Conversely, we can construct the corresponding packing from any sequence-pair along the horizontal and vertical constraint in it. In other words, there always exist a packing for any given sequence-pair (Γ_+, Γ_-) satisfying the horizontal and vertical constraints described above.

The solution space of SEQ-PAIR satisfies the following requirements for effective search algorithms.

(1) The solution space is finite : Each sequence is a permutation of n rectangles. Thus, the size of the solution space is exactly $(n!)^2$.
(2) Every solution is feasible : If a code has its corresponding packing, it is called *feasible* or a feasible solution. As described above, every sequence-pair is feasible.
(3) Evaluation of each code (sequence-pair) is possible in polynomial time and so is the realization of the corresponding packing : For a given sequence-pair, we can calculate the value of the objective function, i.e. the area of corresponding packing, in $O(n^2)$ time[1].
(4) The packing corresponding to the best evaluated code (sequence-pair) in the space coincides with an optimal solution of RP : There exists a sequence-pair corresponding an optimal packing.

A solution space which satisfies the above four conditions is called *P-admissible*. Requirement (2) is for existence of neighbors of every feasible solution, which most heuristics need.

III. LOT(Labeled Ordered Tree)

In this section, we introduce an encoding scheme LOT (the abbreviation for Labeled Ordered Tree) and show that its solution space is also P-admissible.

First, we show how every packing is encoded into a labeled ordered tree.

Let Π be a packing for $S = \{r_1, r_2, \dots, r_n\}$. Without loss of generality, we assume that each rectangle has no room on the left of it in the bounding rectangle, that is, each rectangle touches the right side of some other rectangle or the left side of the bounding rectangle. (If not, we can slide such rectangles without increasing the area of Π.)

The following procedure constructs a labeled ordered tree from Π. To begin with, place a sufficiently large dummy rectangle r_0 on the left side of the bounding rectangle of Π. Each vertex of a labeled ordered tree corresponds to a rectangle. So, we often identify rectangle r_i with a vertex with label r_i.

Packing to Tree

```
Mark $r_0$ and push it into STACK.
While (STACK $\neq \emptyset$) do {
    $r_i \leftarrow$ pop().
    If $r_i$ has an unmarked rectangle $r_j$ on the right, let
        $r_j$ be a son of $r_i$, mark $r_i$, and push $r_i$ into
        STACK. (If we have two or more adjacent rect-
        angles, choose the lowest one as $r_j$.)
}
```

Note that the procedure above is just the DFS (Depth First Search) algorithm. We show an example of a labeled ordered tree in Fig.2.

Conversely, we can construct a packing of S from a labeled ordered tree T whose root has label r_0:

Tree to Packing
According to the preorder of T, 'drop' each corresponding rectangle r_i one by one into a sufficiently large bin so that the left side of r_i and the right side of its parent rectangle are on the same vertical line. The root rectangle r_0, which does not have a parent, is initially placed on the left of the bin.

To clarify the dropping procedure, we show an example of dropping for another tree in Fig.3. Note that we can

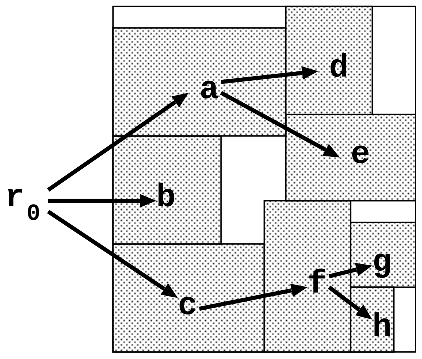

Fig. 2. Construction of a labeled ordered tree

[1]In fact, we can evaluate a code (sequence-pair) in $O(n \log n)$ time[3] .

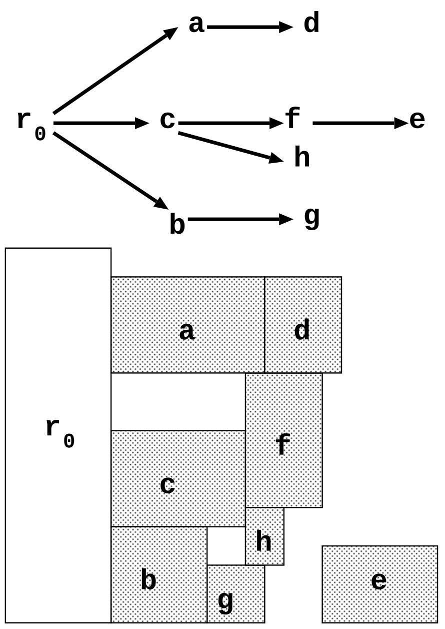

Fig. 3. A labeled ordered tree and the packing given by the dropping procedure

not always get a packing such that each rectangle touches the right side of its parent rectangle. (In Fig.3, rectangle e does not touch its parent rectangle f.)

From the above observations, LOT also satisfies the following properties.

(1) The solution space is finite : The solution space is the set of all labeled ordered trees with $n+1$ vertices whose roots are labeled r_0. (See Section IV.)

(2) Every solution is feasible : Every labeled ordered tree with root r_0 has its corresponding packing.

(3) Evaluation of each code (labeled ordered tree) is possible in polynomial time and so is the realization of the corresponding packing : Construction of the corresponding packing can be computed by the dropping procedure in polynomial time. (See Section V.)

(4) The packing corresponding to the best evaluated code (labeled ordered tree) in the space coincides with an optimal solution of RP :

Thus, the solution space of LOT is also *P-admissible.*

IV. The size of the solution space

How large is the solution space of LOT ? The size of the solution space, say $\mathcal{LOT}(n)$, is the number of labeled ordered trees with $n+1$ vertices whose roots are labeled r_0. The number of ordered tree with n vertices is given by *Catalan number*

$$C_n = \frac{1}{2n+1}\binom{2n+1}{n}.$$

(The reader can refer to many textbooks of combinatorics about Catalan number.)

Thus, $\mathcal{LOT}(n) = c_n \cdot (n!)$ since the root must be labeled r_0. Remind that the size of the solution space of SEQ-PAIR, say $\mathcal{SEQ}(n)$, is $(n!)^2$.

Comparing two solution spaces, we have

$$\lim_{n\to\infty} \frac{\mathcal{LOT}(n)}{\mathcal{SEQ}(n)} = \lim_{n\to\infty} \frac{c_n}{n!} = 0.$$

For the readers' information, we give $n!$ and Catalan number c_n for $n \leq 12$ in Table I.

TABLE I
c_n AND $n!$

n	$n!$	c_n
1	1	1
2	2	2
3	6	5
4	24	14
5	120	42
6	720	132
7	5,040	429
8	40,320	1430
9	362,880	4862
10	3,628,800	16,796
11	39,916,800	58,786
12	479,001,600	208,012

V. Representation of Trees

We have introduced an encoding scheme LOT, which has a small and P-admissible solution space. However, we have not given the data structure to represent the labeled ordered trees yet; The reason is that we have several options.

When we adopt LOT in heuristic search algorithms, we often need to represent labeled ordered trees in strings,for example, in GA. Here, we represent a labeled ordered tree as a string of balanced labeled parentheses.

It its known that there exist one-to-one correspondence between (unlabled) ordered trees and balanced parentheses(Fig.4). Thus, the labeled ordered tree in Fig.2 can be represented as

$$(c(f(hh)(gg)f)c)(bb)(a(ee)(dd)a).$$

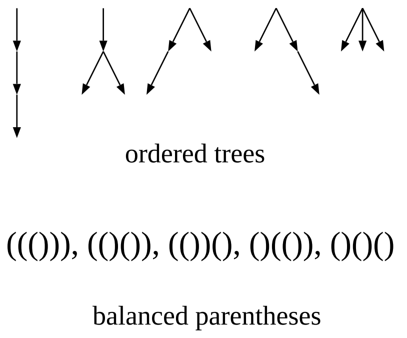

Fig. 4. Ordered trees and balanced parentheses

VI. Time for evaluating a code

Consider constructing a packing by dropping procedure described in Section III. Suppose that a labeled ordered tree is represented as a balanced labeled parentheses in a queue. Then, the following procedure gives the corresponding packing:

```
$r = r_0$
While (QUEUE $\neq \emptyset$) do {
    Get $s$ from QUEUE.
    If $s =$ '$(r_i$' then drop $r_i$ on the right of $r$ and $r \leftarrow r_i$.
    If $s =$ '$r_i)$' then $r \leftarrow$ parent of $r_i$.
}
```

This procedure is essentially same as the traverse of a tree. Keeping track of the width and height of the packing, the procedure gives the area of the packing. Using a doubly linked list of heights of rectangles for the rectilinear envelope of the packing, the procedure calculates the area of packing in $O(n)$ time. Remind that the time complexity of evaluating a sequence-pair is $O(n^2)$[2] or $O(n \log n)$[3].

VII. Conclusion

We have introduced a new encoding scheme LOT. The solution space of LOT is P-admissible as that of SEQ-PIAR. Moreover, it is smaller and the corresponding packings can be computed faster.

However, we should note that the size of the solution space itself is not important; LOT might discard many quasi-optimal solutions to get the small space. We need some experimental results to insist that LOT is a good encoding scheme for heuristic search algorithms.

For the readers' information, we report that a straightforward exhaustive search algorithm gave an optimal solution for seven rectangles in several minutes (Fig.5). By 'straightforward exhaustive search,' I mean that the program generated and evaluated 2,162,160 labeled balanced parentheses.

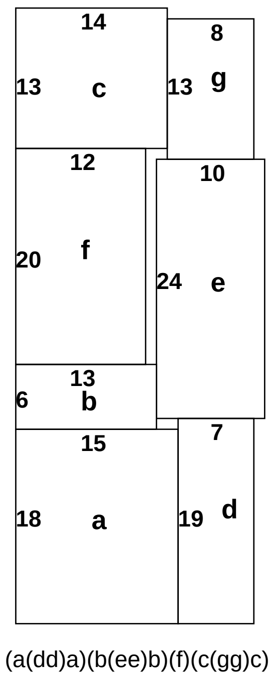

Fig. 5. An optimal packing ($n = 7$)

Acknowledgements

This work was supported by Research Body CAD21, Tokyo Institute of Technology.

References

[1] B. S. Baker, E. G. Coffman, and R. L. Rivest, "Orthogonal Packings in Two Dimensions," *SIAM J. Compute.*, Vol. 9, No. 4 , pp. 846-855, 1980.

[2] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "Rectangle-Packing-Based Module Placement," *Proc. IEEE/ACM International Conf. on Computer-Aided Design*, pp. 472-479, 1995.

[3] T. Takahashi, "An Algorithm for Finding a Maximum Weight Decreasing Sequence in a Permutation Motivated by Rectangle Packing Problem," *Technical Report IEICE*, Vol. VLD96, No. 201, pp. 31-35, 1996.