３次元箱詰め問題に対する実用的アルゴリズムの開発

今　堀　慎　治*

Practical algorithms for three-dimensional packing problem

Shinji IMAHORI*

In this report, practical algorithms for cutting and packing problems are studied. We first consider a problem of enumerating bottom-left stable positions for a given layout of rectangles. We introduce an efficient algorithm that enumerates all the bottom-left stable positions in $O((n + K) \log n)$ time, where n is the number of placed rectangles (i.e., input size) and K is the number of bottom-left stable positions (i.e., output size). This algorithm works for layouts without bottom-left stability and with overlaps. An important consequence of this algorithm is that it can be utilized to design an efficient algorithm to execute a bottom-left algorithm for three-dimensional packing problem. We show that the time complexity is improved from the previous best-known $O(m^5)$ to $O(m^3 \log m)$ to place m rectangular solids.

1. Introduction

Cutting and packing problems are important problems with applications in various industries such as steel, wood, glass and paper. There are many variants of the problem with different objectives and constraints, but the essential task is to place a given set of items in a given larger area without overlap so that the wasted space in the resulting layout is minimized. Almost all variants of the problem are known to be NP-hard, and many heuristic algorithms have been proposed in the literature. One of the typical frameworks of existing heuristic algorithms is the bottom-left strategy, which places items one by one at bottom-left stable positions[1,2]. A fundamental problem to be solved for executing these algorithms is to enumerate all bottom-left stable positions for a set of already placed items and one new item to be placed next.

Bottom-left stable positions are defined for a given area, a set of items placed in the area, and one new item. It is a point in the area where the new item can be placed without overlaps with already placed items and the new item cannot move to the bottom or to the left. We also define bottom-left stability for a layout; if there is no overlap among items and no item can move to the bottom or to the left, the layout satisfies bottom-left stability.

In this report, we consider the problem of enumerating bottom-left stable positions for a new rectangle within a given layout of rectangles that may not satisfy bottom-left stability and may have overlaps between rectangles. We introduce an enumeration algorithm that runs in $O((n + K) \log n)$ ctime, where n is the number of placed rectangles and K is the number of bottom-left stable positions.

The bottom-left strategy can naturally be generalized to the three-dimensional case. An important consequence of our enumeration algorithm is that it can be utilized to design an efficient bottom-left algorithm for the three-dimensional packing problem; that is, existing algorithms such as those proposed by Chazelle[2] and Healy[3] cannot be used for this purpose. We show that the time complexity is improved from the previous best-known $O(n^5)$ to $O(n^3 \log n)$.

2. Problem description

In this section, we introduce two problems treated in this report. The first one is the enumeration problem of the bottom-left stable positions for a given layout of rectangles. We are given a set of n rectangles $I = \{1, 2, \cdots, n\}$ and one large rectangular area, also called the container. Each rectangle $i \in I$ has its width and height (w_i, h_i), and is placed orthogonally in the plane. Let (x_i, y_i) be the coordinate of the bottom left point of rectangle i. We note that the given rectangles may have overlaps. The container has its width and height (W, H) and its bottom left point is placed at $(0, 0)$ in the plane. We are also given one new rectangle $J \notin I$ with size (w_j, h_j) that has not been placed in the area yet. The objective is to enumerate all the bottom-left stable positions

2011年1月31日　受理

* 豊田理化学研究所研究嘱託（名古屋大学大学院工学研究科）

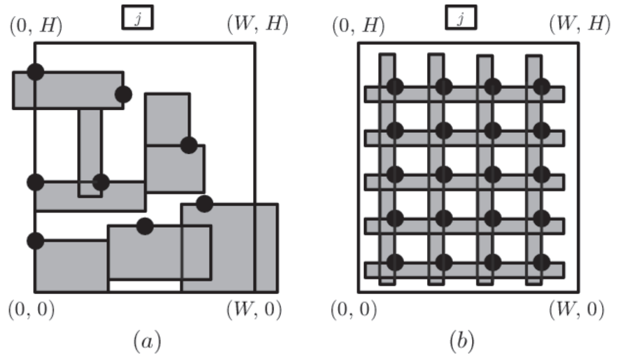

Fig. 1. Bottom-left stable positions for a rectangle.

in the container for rectangle j. See Figure 1 for an example of bottom-left stable positions; black points in this figure denote bottom-left stable positions for rectangle j. Let K be the number of bottom-left stable positions for a given layout and one new rectangle. It is known that $K = \mathrm{O}(n^2)$ and K can be $\Theta(n^2)$ for some cases (see Figure 1(b) for an example).

We then explain the three-dimensional packing problem. We are given a set of m rectangular solids and one large container with fixed width W, fixed height H and variable depth D. The objective is to place all the rectangular solids in the container without overlaps so as to minimize the variable depth.

3. Algorithms

We explain algorithms to enumerate bottom-left stable positions for a given layout of rectangles and its application. We first introduce no-fit polygon, which is often used in packing algorithms to check overlaps efficiently. We then explain a technique to compute for each point p in the plane the number of no-fit polygons containing p by using sweep line. In Section 3, we show an algorithm for enumerating bottom-left stable positions for a rectangle. At the end of this section, we mention the three-dimensional packing problem.

For the first problem, instead of considering the constraint that requires a new rectangle to be placed in the container, we use the set of four sufficiently large virtual rectangles $C = \{c_1, c_r, c_t, c_b\}$ (called container rectangles) that satisfies the following condition: Rectangle j does not have overlaps with rectangles $i' \in I \cup C$ if and only if it is placed in the container without overlaps with rectangles $i \in I$. We denote $I' = I \cup C$; then $|I'| = |I| + 4$ holds.

3.1. No-fit polygon

No-fit polygon is a geometric technique to check overlaps of two polygons in two-dimensional space. It is defined for an ordered pair of two polygons i and j, where the position of polygon i is fixed and polygon j can be moved. $NFP(i,j)$ denotes positions of polygon j having intersection with polygon i. Let rectangle i be placed at (x_i, y_i) and rectangle j be the new rectangle. Then $NFP(i,j)$ is defined as follows:

$$NFP(i,j) = \{(x,y) \mid x_i - w_i < x < x_i + w_i,\ y_i - h_i < y < y_i + h_i\}.$$

We also define the overlap number $B(x, y)$ of no-fit polygons at point (x, y) as follows:

$$B(x,y) = |\{i \in \mathrm{I}' \mid (x,y) \in NFP(i,j)\}|.$$

By using this overlap number, we can characterize bottom-left stable positions as follows:

$$(x, y) \text{ is a bottom-left stable position} \Leftrightarrow$$
$$\mathrm{B}(x, y) = 0 \wedge \mathrm{B}(x - \varepsilon, y) > 0 \wedge \mathrm{B}(x, y - \varepsilon) > 0,$$

where ε is any sufficiently small positive number. In the next section, we will describe how to compute overlap numbers of no-fit polygons.

3.2. Compute overlap numbers

The algorithm first computes all no-fit polygons $NFP(i,j)$ of rectangle j relative to placed and container rectangles $i \in \mathrm{I}'$. In order to compute overlap numbers of no-fit polygons in the given area efficiently, the algorithm uses a sweep line parallel to the x-axis and moves it from bottom to top.

Let N_t (resp., N_b) be the set of all the top (resp., bottom) edges of no-fit polygons and $N_{tb} = N_t \cup N_b$. The overlap numbers on the sweep line will be changed only when the sweep line encounters a member of N_{tb}, and changes occur only in the interval between the left edge and right edge of the no-fit polygon encountered by the sweep line.

Let N_1 (resp., N_r) be the set of all the left (resp., right) edges of no-fit polygons and $N_{1r} = N_1 \cup N_r$. Because there are n placed rectangles and four container rectangles, $|N_t| = |N_b| = |N_1| = |N_r| = n + 4$ and $|N_{tb}| = |N_{1r}| = 2n + 8$ hold. The elements in N_{1r} are sorted in nondecreasing order of the x-coordinates of the elements. Let $x_{1r}^{(k)}$ be the x-coordinate of the kth element in the sorted list of N_{1r}, and define intervals

$$S_k = \left[x_{1r}^{(k)}, x_{1r}^{(k+1)}\right],\ k = 1, 2, \cdots, 2n + 7$$

on the sweep line.

The algorithm maintains the overlap number for each interval S_k during the computation. Initially, the sweep line is at a sufficiently low position, and it does not overlap with no-fit polygons. At this moment, the overlap number of every interval S_k is zero.

We now consider the moment when the sweep line encounters a member in N_{tb}. Let $NFP(i,j)$ be the rectangle whose top or bottom edge is encountered by the sweep line, and assume that the left (resp., right) edge of $NFP(i,j)$ is the lth (resp., $(r+1)$st) element in the sorted list of N_{1r}. In this situation, we should change the overlap numbers for intervals S_l, S_{l+1}, ·····, S_r. To be more precise, we should increase (resp., decrease) their overlap numbers by one if the encountered edge is a member of N_b (resp., N_l). To update overlap numbers on the sweep line efficiently, we use a complete binary tree whose leaves represent intervals S_1, S_2, ·····, S_{2n+7}. Due to the limitation of pages, we omit the details of the complete binary tree and denote only the results (see the paper by Imahori *et al.*[4] for the details). It is possible to compute the overlap number of an interval in O(log n) time. It is also possible efficiently to check whether intervals whose overlap numbers are equal to zero exist in some consecutive intervals.

3.3. Enumerate bottom-left stable positions

We explain our algorithm that enumerates bottom-left stable positions. Observe that, while the sweep line parallel to the x-axis is moved from bottom to top, the overlap numbers of no-fit polygons for intervals in the sweep line decrease only if the top edge of a no-fit polygon is encountered. This means that bottom-left stable positions can be found only in this case, because a point (x, y) can be a bottom-left stable position only if $B(x, y) = 0$ and $B(x, y - \varepsilon) > 0$ for any sufficiently small positive ε. For this reason, when the sweep line encounters the bottom edge of a no-fit polygon, the algorithm just updates the overlap numbers. On the other hand, when the sweep line encounters the top edge of a no-fit polygon, the algorithm updates the overlap numbers and outputs bottom-left stable positions on the sweep line if such positions exist. To manage these events, the elements in N_{tb} are sorted in nondecreasing order of the y-coordinates of the elements.

At any point (x, y) such that the overlap number $B(x, y)$ is equal to zero, we can place rectangle j without overlap. Moreover, if the overlap number becomes zero when the top edge of a no-fit polygon is encountered by the sweep line, then $B(x, y - \varepsilon) > 0$ for any sufficiently small $\varepsilon > 0$, i.e., rectangle j cannot move downward from the point. Furthermore, if the point (x, y) is at the left boundary of an interval S_k and its left adjacent interval S_{k-1} has a positive overlap number, then $B(x - \varepsilon, y) > 0$ for any sufficiently small $\varepsilon > 0$, i.e., rectangle j cannot move to the left. Such a point (x, y) is a bottom-left stable position, and our algorithm enumerates all such points in O(log n) time per one such position.

3.4. Three-dimensional packing

The bottom-left strategy can naturally be generalized to the three-dimensional problem in theory, but its fast implementation has not been known; the previous best implementation requires $O(m^5)$. By using our enumeration algorithm for rectangles as a core part, we design a bottom-left algorithm for the three-dimensional packing problem, which utilizes the sweep-plane technique and runs in $O(m^3 \log m)$ time. See the paper by Kawashima *et al.*[5] for the details.

4. Conclusions

Cutting and packing problems are important in many industries and practical algorithms for these problems are needed. In this report, we considered the problem of enumerating bottom-left stable positions for a given layout of rectangles. Our algorithm for this problem runs fast and works for layouts without bottom-left stability and with overlaps. By using this algorithm as a core part, we designed a practical algorithm for the three-dimensional packing problem, which places m rectangular solids in $O(m^3 \log m)$ time.

References

1) B.S. Baker, E.G. Coffman, R.L. Rivest, Orthogonal packings in two dimensions. *SIAM Journal on Computing.* **9** (1980) 846-855.
2) B. Chazelle, The bottom-left bin packing heuristic: an efficient implementation. *IEEE Transactions on Computers.* **32** (1983) 697-707.
3) P. Healy, M. Creavin, A. Kuusik, An optimal algorithm for rectangle placement. *Operations Research Letters.* **24** (1999) 73-80.
4) S. Imahori, Y. Chien, Y. Tanaka, M. Yagiura, Enumerating bottom-left stable positions for rectangles with overlap. *Proceedings of the 9th Forum on Information Technology 2010 (FIT2010)*, Issue 1, pp.25-30.
5) H. Kawashima, Y. Tanaka, S. Imahori, M. Yagiura, An efficient implementation of a constructive algorithm for the three-dimensional packing problem. *Proceedings of the 9th Forum on Information Technology 2010 (FIT2010),* Issue 1, pp.31-38.