

# A NEW HEURISTIC ALGORITHM FOR TWO-DIMENSIONAL DEFECTIVE STOCK GUILLOTINE CUTTING STOCK PROBLEM WITH MULTIPLE STOCK SIZES

*Maozhu Jin, Pen Ge, Peiyu Ren*

Original scientific papers

This paper mainly addresses a two-dimensional defective stocks guillotine cutting stock problem where stock of different sizes is available. Herein a new heuristic algorithm which is based on tree is proposed to discuss this problem. In particular, such an algorithm consists of two parts: the first part is an initial solution of the cutting stock problem where there are no defects on the stocks; the second part is the final optimization solution which is set up on the basis of the first part and takes the defects into consideration. This paper also evaluates the performance of the proposed algorithm. The experimental results demonstrate the effectiveness of the algorithm for the two-dimensional defective stocks cutting stock problem and show that the algorithm can improve not only the utilization rate of stocks, but also the reuse rate of remainders by reducing the fragmentation of remainders.

**Keywords:** *combinatorial optimization; cutting and packing; defects; heuristics*

## Novi heuristički algoritam za problem rezanja giljotinom dvodimenzijске oštećene robe s višestrukom veličinom

Izvorni znanstveni članak

U radu se uglavnom raspravlja o problemu rezanja giljotinom dvodimenzijске oštećene robe raspoložive u različitim veličinama. Za raspravu o problemu predlaže se novi heuristički algoritam u obliku stabla. Takav se algoritam sastoji od dva dijela: prvi dio je početno rješenje problema rezanja robe kad ne postoje oštećenja robe; drugi dio je konačno rješenje optimizacije utemeljeno na prvom dijelu uz razmatranje oštećenja. U radu se također ocjenjuju rezultati predloženog algoritma. Eksperimentalnim se rezultatima demonstrira učinkovitost algoritma za problem rezanja dvodimenzijске oštećene robe i pokazuje da se algoritmom može poboljšati ne samo stopa iskoristivosti robe već i stopa ponovne uporabe ostataka smanjenjem fragmentacije ostataka.

**Кljučне riječi:** *heuristika; kombinatorna optimizacija; oštećenja; rezanje i pakiranje*

## 1 Introduction

Given a set of rectangular items and several identical rectangular stocks, the two-dimensional cutting stock problem (2DCSP) requires cutting all the items by using the minimum number of stocks or, equivalently, by minimizing the global area of the used stocks. If the stocks are not identical, it is a 2DCSP with multiple stock sizes (MS2DCSP). However, in the rolling process, there will be some defects in the stocks, such as chinks, knot-holes, shrinkage and so on. Therefore, 2DCSP and MS2DCSP are often not suitable in practical applications. This paper considers multiple stock sizes into the two-dimensional defective stocks guillotine cutting stock problem (MS2DDSGCSP), that is, 2DCSP where: (1) Only guillotine cuts are allowed, i. e. cuts that run from one edge of a rectangle to the opposite edge, parallel to the other two edges. (2) Stocks of different sizes are available. (3) Items can be orthogonally rotated. (4) Some stocks have defects. There may be some defects in the stocks and items should avoid the defects. 2DCSP, MS2DCSP and MS2DDSGCSP are all NP-hard since the one-dimensional cutting stock problem, which is a special case of these problems, is NP-hard.

2DCSP was introduced by Gilmore and Gomory (1965) [1], Then Lodi et al. (2002) [2] provided an overview of the problem and some of its variants. Several exact methods and lower bounds have been proposed for the problem. Christofides and Whitlock (1977) [3] used tree search method to get an exact algorithm, and put forward the dynamic programming on solving practical problems. Then, more exact algorithms have been proposed for the 2DGC-SP [4÷8]. Further, some exact algorithms have been proposed for the MS2DCSP [9÷12]. However, since the 2DCSP and MS2DCSP are NP-hard,

the exact algorithm needs more time to find a solution, heuristic algorithms are more popular [13] to solve the large problem instances encountered in practice. Various heuristic algorithms based on different methodologies have been presented for solving these problems, such as Best-fit (BF) method and its enhancement [14, 15], Intelligent Search Algorithm and Hybrid Simulated Annealing Algorithm [16, 17], Parallel Algorithm [18], Sequential Grouping Heuristic [19, 20] and so on.

The two-dimensional defective-stock guillotine cutting stock problem (2DDSGCSP) is comparatively less well-studied, although a few related variants have been investigated. Gilmore and Gomory (1965, 1966) [1, 21] developed a fundamental solution strategy based on the revised simplex method with column generation technique. Hahn (1968) [22] was the first one to consider the problem of cutting rectangular pieces out of rectangular defective stocks, then Scheithauer and Terno (1988) [23] improved the solution process, but these studies are insufficient. Later, Twisselmann (1999) [24] put forward an effective algorithm to find all usable restricted rectangles and not the maximum empty rectangle only. More recently, Neidlein et al. (2008) [25] suggested several modifications of the AND/OR-graph approach which allow for dealing with a stock that contains a single defect.

The following notation is utilized in developing the mathematical model of the MS2DDSGCSP addressed in this study:

$(h; w)$ : height and width of the item.

$(H; W)$ : height and width of the stock.

$I$ : number of the stocks with the same size.

$J$ : number of the items with the same size.

$B = \{ \langle W_i, H_i \rangle | i \in I \}$ : set of the stocks.

$A = \{ \langle w_j, h_j \rangle | j \in J \}$ : set of the items.

$D_{lm}$ : the  $m^{\text{th}}$  defect  $l$ .  
 $(w_j, h_j)$ : the size of item  $j$ .  
 $(w_l, h_l)$ : the size of defect  $l$ .  
 $n_j$ : number of item  $j$ .  
 $n_{ij}$ : number of item  $j$  on the stock  $i$ .  
 $(x_{0jk}^i, y_{0jk}^i)$ : top right corner coordinates of the  $k^{\text{th}}$  item  $j$  on the stock  $i$ .  
 $(x_{1jk}^i, y_{1jk}^i)$ : bottom left corner coordinates of the  $k^{\text{th}}$  item  $j$  on the stock  $i$ .  
 $dx$ : the horizontal distance between left bottom corner of item (defect) and right top corner of defect (item).  
 $dy$ : the vertical distance between left bottom corner of item (defect) and right top corner of defect (item).  
 $p_{jk}^i$ : layout pattern of the  $k^{\text{th}}$  item  $j$  on the stock  $i$ .  
 $p_{jk}^i=0$ , Vertical cut; else 1, Horizontal cut.

The literature about 2DDSGCSP only relates to the single-stock or single defect, but in the practical application, these situations are not common. This paper proposes a new heuristic algorithm to investigate the MS2DDSGCSP problem. There are two main contributions made by this paper. Firstly, a Strip and Shake operator is proposed to reduce the fragmentation of remainders, which can improve not only the utilization rate of stocks, but also the reuse rate of remainders. Secondly, the defects in the stocks are taken into consideration and the MS2DDSGCSP where defect of different sizes are present.

The rest part of this paper is organized as follows. Section 2 builds a mathematical model for the MS2DDSGCSP. Section 3 proposes a heuristic algorithm for the solution of the investigated problem. In Section 4, some experimental results are presented. Finally, this paper draws some conclusions and looks forward to the further research in Section 5.

**2 Problem formulation**

In this section, the MS2DDSGCSP is formulated.

**2.1 Constraints**

The actual application has many peculiar characteristics and some constraints, which is described mathematically as follows.

(1) Item size constraints.

$$x_{1jk}^i = x_{0jk}^i + p_{jk}^i w_j + (1 - p_{jk}^i) h_j, \tag{1}$$

$$y_{1jk}^i = y_{0jk}^i + p_{jk}^i h_j + (1 - p_{jk}^i) w_j. \tag{2}$$

(2) Item cross constraints. All the items cannot overlap with the others.

if  $y_{0tr}^i < y_{1jk}^i$ , then  $x_{0tr}^i \geq x_{1jk}^i$   
 $(t, j \in J; r \in n_t; k \in n_j), \tag{3}$

if  $x_{0tr}^i < x_{1jk}^i$ , then  $y_{0tr}^i \geq y_{1jk}^i$   
 $(t, j \in J; r \in n_t; k \in n_j). \tag{4}$

(3) Stock constraints. Items cannot be outside of the stocks.

$$\max_{j,k} \{x_{1jk}^i\} \leq W_i, \tag{5}$$

$$\max_{j,k} \{y_{1jk}^i\} \leq H_i. \tag{6}$$

(4) Item number constraints. The total number of the item  $j$  on all the stocks is equal to  $n_j$ .

$$\sum_{i=1}^I n_j^i = n_j. \tag{7}$$

(5) Defect constraints. All the items cannot overlap with the defects.

$$dx > w_j + w_1, \tag{8}$$

$$dy > h_j + h_1. \tag{9}$$

**2.2 Objective function**

The main purpose of solving the cutting stock problem is to maximize the utilization of stocks. This paper tries to minimize the maximum rectangular envelope, while maximum rectangular envelope is the minimum rectangular containing all the items on the stock, which can be expressed as follows:

$$\min \sum_{i=1}^I \max_{j,k} \{x_{1jk}^i\} * \max_{j,k} \{y_{1jk}^i\}. \tag{10}$$

**3 Heuristic algorithm**

An effective heuristic algorithm for MS2DDSGCSP is proposed in this section, which can be divided into two parts. The first part mainly researches the 2DGCSP, while the second part gains an initial solution on the basis of the first part and gets the final solution at last considering and dealing with the defects in the stock. A flow chart is presented to express the process of the algorithm.

**3.1 Flow chart for the algorithm**

The heuristic algorithm can be shown in the flow chart (as shown in Fig. 3). The first part, consisting of the first three steps, is applied to investigate 2DGCSP. The second part is composed of the latter two steps, mainly dealing with the defects:

Step 1, the choice of the stock. According to the largest edge, sequence the optional stocks in ascending order and the waiting-pack-items in descending order, and for the selected biggest items, select available minimum stock from the stock library to pack. If the size of stock is not enough, choose another stock.

Step 2, initial packing solution. According to the rules of minimum height and bottom left, a feasible packing solution of the elected stock can be obtained. When inserting an item, we can find out the first type cavity for

merging and get the initial packing solution until there is no waiting-pack-items or items that could be inserted.

Step 3, Dynamic optimization. The special second type cavity found in the initial packing solution will be merged. Then Strip and Shake operator is used to gather and move the cavity to the top right-hand corner of the stock and find the item that can be inserted into the cavities from the waiting-pack-items. If it exists, insert the item; if there are no waiting-pack-items or the merged cavities could not be inserted with any items, keep on searching for the adjacent cavity to merge until it cannot satisfy the conditions of merging, then getting the optimization packing solution on a single stock.

Step 4, Eliminate the items that overlap with the defects. This paper searches defects on the whole stock

and judges whether the items overlap with the defects. Once overlapping with the defects, the items will be eliminated immediately. Repeat the operation until no item overlaps with the defects

Step 5, Boom operators. This paper designs a boom operator for defects to make the items as far away from the defects as possible, searches the boundary of the defect, gathers the cavities and merges the adjacent cavities, then inserting new items.

Return the remainders produced in the above packing solution to the alternative stock library. Start the next round of the packing from part 1 until all the waiting-cut-items have been inserted completely.

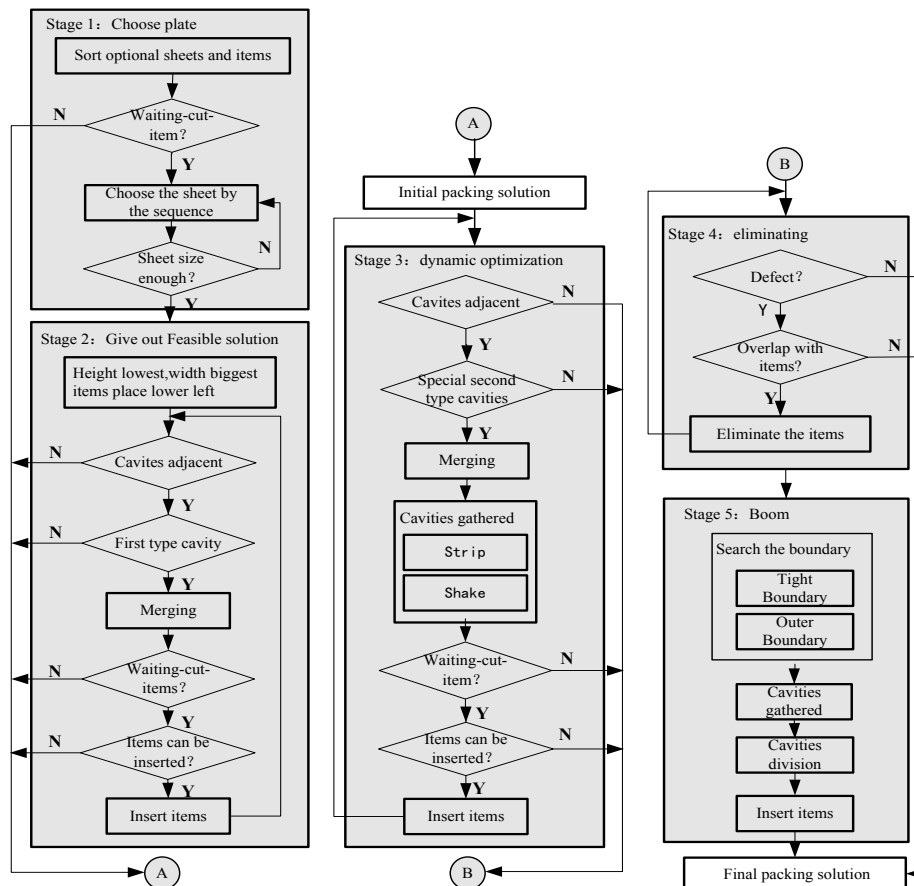


Figure 1 Dynamic optimization algorithm flow chart

### 3.2 First part: A binary tree-based heuristic algorithm

This section proposes a binary tree-based heuristic algorithm for 2DGC-SP. The related cutting rule is firstly introduced in section 3.2.1. In section 3.2.2, binary tree optimization idea is applied to express guillotine cut. Finally in section 3.2.3, a series of packing operators are proposed.

#### 3.2.1 Basic rules for cutting

*Insert rule:* Minimum cavities for biggest items.

- Big edge: the bigger edge between the height and the width of the stocks (or items).
- Biggest items: the waiting-pack-items with maximum big edge.

- Minimum cavities: the waiting-cut-stocks with minimum big edge.

*Insert rule* is used to choose stocks. Sequence the waiting-pack-items in descending order according to the big edge; sequence the waiting-cut-stocks in ascending order according to the big edge of the stock. Then set that, in a new round packing process, the biggest items of the waiting-pack-items will be chosen. Search the cavity list, and choose the minimum cavity from the cavities that satisfy the item size as the stocks. Therefore, the heuristic algorithm proposed in this paper will be compatible with multi-stocks cutting problems.

*Place rule:* bottom left for a given cavity in initial cutting scheme, the maximum width and the minimum height waiting-pack-item will be chosen, and placed on bottom left.

**Rotation rule:** Modulus method set the small edge of the cavity as  $x = \min(W, H)$ , where  $W$  and  $H$  are the width and the height of the cavity respectively. Denote  $w, h$  as the width and height of the item which is chosen by insert rule and place rule. So the  $x, w$  and  $h$  exist in the following linear relationship:  $x = a \cdot w + d_1; x = b \cdot h + d_2$ . If  $d_1 < d_2$ . Place the item as  $w||x$ , or as  $w||x(w||x$  means that the width of the item is parallel to the small side of the cavity).

**Cutting rule:** Integral point cutting.

**Definition 1.** Connected cavity: If there are two or more mergeable second type cavities (see section 3.2.3) in the packing solution, remove the public edge between the cavities to compose a cavity that is not cut, and this cavity is called connected cavity (shown in Figs. 6b ÷ 6c).

**Definition 2.** Integral point coordinates: A set of coordinates is composed of the Cartesian product for the bottom left coordinates and top right coordinates of the items that have been all packed, and they are all located in connected cavity.

In order to meet the guillotine cut constraint, the bottom left coordinates of the items which are used to fill the connected cavity should be located on the integral point coordinates, and the size of the items that have been chosen should maximize the utilization rate of the connected cavity.

### 3.2.2 Binary tree optimization

We use the generation process of binary tree to describe the segmentation process of the stock, and then the generation process of binary tree stands a packing result, means getting a global packing solution.

**Definition 3.** Local solution:

Denote  $I = \langle N, G \rangle | N = \{n_j\}; G = \{g_j\}, j = 1; 2 \dots \}$  as a global packing solution, where  $N$  stands for the set of all kinds items' number,  $G$  stands for a set of the items' size.  $\exists i = \langle N_i, G_i \rangle | N_i \subseteq N, G_i \subseteq G$  is local area of  $I$ , so call  $i$  as a local solution of  $I$ .

The characteristics of guillotine cut constraints endow the packing solution with the following properties:

**Property 1:** The first cutting line in the packing solution has to ensure that the length is equal to its parallel edges length in the segmented regions and does not affect integrity of any items.

**Property 2:** Denote  $I$  as the global solution that meets guillotine cuts. For any local solution  $i$  in  $I$ ,  $i$  must satisfy guillotine cuts.

**Theorem 1.** Cutting line replacement theorem: For any feasible solutions, if there exists another cutting line that satisfies the cutting condition of the first cutting line, the solution formed by taking this cutting line as the first cutting is the same as the original solution.

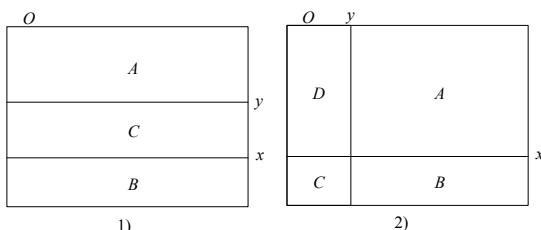


Figure 2 Two cases for Cutting line replacement theorem

**Proof:** As shown in Fig. 2, have known area O exist a feasible solution  $I_x$  that takes  $x$  as the first cutting line, according to Property 2, there exists the local area  $o$  in O to make the solution of taking  $x$  as the first cutting line  $i_{ox} \subseteq I$ . Another cutting line  $y$  has been found now which meets Property 1, and it needs to prove  $I_y = I_x$ . At this time, cutting line  $y$  has two cases, one is  $y$ , parallel to  $x$ , the other is  $y$ , vertical to  $x$ . The two cases are proved below:

1)  $y||x$ , O is divided into  $\{AUC\}$ , B two pieces by  $x$ , and is divided into A,  $\{BUC\}$  two pieces by  $y$ . From Property 2, there is  $i_{\{AUC\}} \subseteq I$  and  $i_B \subseteq I$ , similarly,  $i_A \subseteq I_{\{AUC\}}$  and  $i_C \subseteq I_{\{AUC\}}$ , according to the transitivity of the set, there is  $i_A \subseteq I$  and  $i_C \subseteq I$ . So the packing scheme formed by taking  $y$  as the first cutting line is the same with original solution.

2)  $y \perp x$ , O is divided into  $\{AUD\}$ ,  $\{BUC\}$  two pieces by  $x$ , and is divided into  $\{AUD\}$ ,  $\{AUB\}$  two pieces by  $y$ . From Property 2, there is  $i_{\{AUD\}} \subseteq I$  and  $i_{\{BUC\}} \subseteq I$ , similarly,  $i_A \subseteq i_{\{AUD\}}$  and  $i_D \subseteq i_{\{AUD\}}$ ,  $i_B \subseteq i_{\{BUC\}}$  and  $i_C \subseteq i_{\{BUC\}}$ , according to the transitivity of the set, there is  $i_A \subseteq I, i_B \subseteq I, i_C \subseteq I, i_D \subseteq I$ . So the packing scheme formed by taking  $y$  as the first cutting line is the same with original solution.

Cutting line replacement theorem is proved.

### 3.2.3 Packing optimization operators

This paper defines a series of operators for the following purposes:

(1) To integrate the remainders, and improve the reuse utilization ratio of the stock.

(2) To get the optimum initial packing solution quickly. These operators reduce the number of iterations and improve the speed of searching solution for the entire algorithm, which constitutes the core of the proposed algorithm.

**Split operator.** In this paper, cutting stock is called split, binary tree is adopted to express the stock cutting scheme, and every guillotine cut can be expressed as the following. Parent node  $AB =$  child node A + child node B. The height of AB is  $h_{AB}$  and width  $w_{AB}$ . If child nodes A and B are arranged in the parent node in different ways, the split is also different, the variable

$$y_{node} = \begin{cases} 1, & \text{Vertical cut} \\ 0, & \text{Horizontal cut} \end{cases}$$

is used to represent the way of split, which can be described as follows:

**Horizontal cut:**  $y_{node} = 0$  (as shown in Fig. 3a). According to maximum big edge items selecting stock, when the big edges of the items are equal, smallest height item will be placed lower. There is no impact on the final analysis. Thus, there exist:  $h_{AB} = h_A + h_B; w_{AB} = \max\{w_A, w_B\}$  (Horizontal cut priority is the default cut way in feasible solution).

**Vertical cut:**  $y_{node} = 1$  (as shown in Fig. 3b). According to maximum big edge items selecting stock, when the big edges of the items are equal, the maximum width item will be placed on the left. There is no impact on the final

analysis. Thus, there exist:  $h_{AB} = \max\{h_A, h_B\}$ ;  $w_{AB} = w_A + w_B$ .

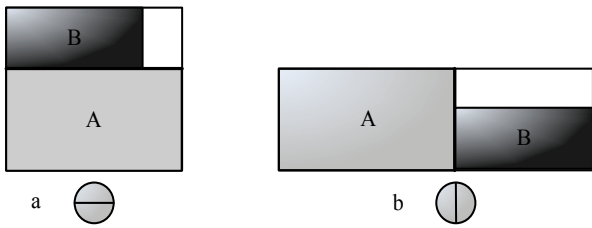


Figure 3 Split: a) Horizontal cut, b) Vertical cut

*Merging operator*

**Definition 4.** Special second type cavity: the public edge, belonging to second type cavity, is much larger than the other edge.

An initial feasible solution can be obtained through the Split operator, and then the adjacent cavities in the initial feasible solution are classified in three categories: The first category has full-scale public edge (as Fig. 4a A and O, B and O, C and O show), which is called first type cavity. The second category has part public edge (as Fig. 4b A and B, B and C, A and C show) and is called second type cavity. The two types of cavities are side contacts, both of which can be merged. The third category belongs to the point contact cavity (as Fig. 4a A and B, B and C show), and it cannot be merged.

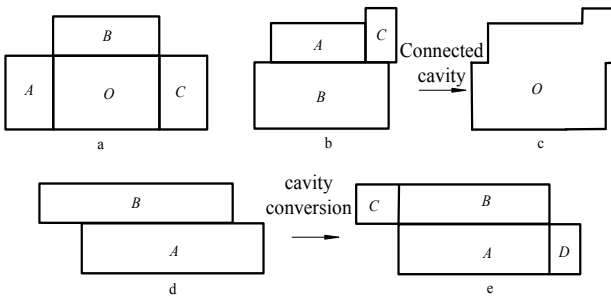


Figure 4 Adjacent cavity

The merging operator of this article is mainly used to merge first type cavity, or, for the special second type cavity (as Fig. 4d A and B show), it needs to be converted into first type cavity (the conversion process shown in Figs. 6d ÷ 6e) and then merge. Record the mergeable cavity pair as  $O = \{(O_i, O_j), i, j \in N_+ \text{ and } i \neq j\}$ .

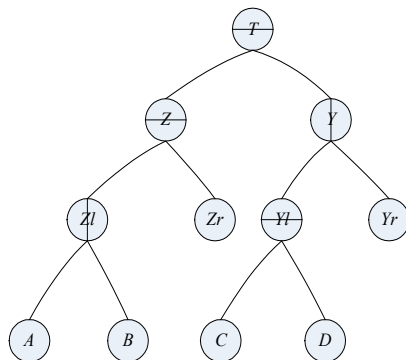


Figure 5 Binary tree representation

**Definition 5.** The lowest common ancestor: If the  $\forall$  cavities are exactly located in the queue of a cavity's left

and right child nodes, then call the cavity as the cavity pair's lowest common ancestor.

Use  $H_{\text{node-node}}$  as the relative depth between the nodes, set the lowest common ancestor of the cavity as Z, left node of Z as Zl, right node Zr.  $y_z, y_{zr}, y_{zl}$  present as cut method,  $H_{Zr-Z}, H_{Zl-Z}$ , respectively represent depth of Zr and Zl related to Z. In Fig. 6,  $y_z = 0$ , but  $y_Y = 1$ , so the relative depth of node B to Z is as follows:  $H_{B-Z} = H_{B-Zl} + H_{Zl-Z} = 2$ , while the relative depth of node Zr to C can be expressed as follows:  $H_{Zr-C} = H_{Zr-Y} + H_{C-Y} = 5$ .

In order to perform merging operator, each mergeable cavity pair (as the cavity A and cavity B) has four states (as shown in Fig. 6a). Starting from each state, within its lowest common ancestor (see Fig. 6 the left part of straight line o), detect the segmentation path and search the replaceable cutting line constantly until the mergeable cavity pair itself becomes the lowest common ancestor. According to the Cutting line replacement theorem, the packing solution formed by replacing the cutting line is the same with the original scheme.

For example, the merging operator can be analyzed as follow: In order to find a mergeable cavity pair (A, B) in Z, from Fig. 6.a that can be seen, the straight line x is the first cutting line of its lowest common ancestor, from four direction search, find  $y_{11}$  ( $y_{11}$  indicates the first replaceable cutting line searched from state 1 to (shown in Figure 6.b), it replaces x and becomes the first cutting line of the lowest common ancestor, then the lowest common ancestor becomes the rectangle enveloped by straight line o,  $y_{11}$ , x. Repeat the above steps, the second time find  $x_{41}$  from the state 4 (Figure 6.c), replace  $y_{11}$  and it is needed to redefine the lowest common ancestor; third from the state 2 to find  $y_{21}$  (Figure 6.d), and so on, until the replaceable cutting line that has been found can completely envelope the cavity pair (A,B), then, merge (A,B) as AB. At this point, one time cavities merge has been completed.

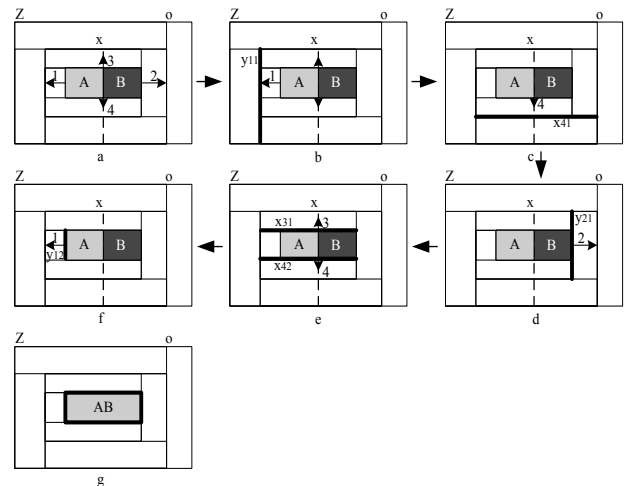


Figure 6 Cavity merging process

**Strip operator Horizontal Strip:** At this point, the cavity should be at the 3,4 state in Fig. 6a, do the merging operator along their direction, the items that have been arranged are divided into horizontal strips (as shown in Fig. 7b, the areas divided by the horizontal line  $x_1, x_2, x_3$ ).

**Vertical Strip:** At this point, the cavity should be at 1,2 state in Fig. 6a, do the merging operator along the x

direction, the items have been arranged and are divided into vertical strips (as shown in Fig. 7d, the areas divided by the vertical line  $y_1, y_2$ ).

The initial packing solution formed through performing split and merging operator has some unreasonable cavities. For example, in Fig. 7a, cavity A is an unreasonable cavity, which is mainly determined by the first cutting line  $x$ . In order to eliminate the A, the

Shake operator should be done for its upper items to make the items drop down. However, before doing Shake operator, the relative depth of A with its upper items should be reduced to 2. At this time, Strip operator is necessary to be done.

Strip operator makes the items that have been packed orderly in horizontal and vertical directions change the cutting order and generates no effect on cutting results.

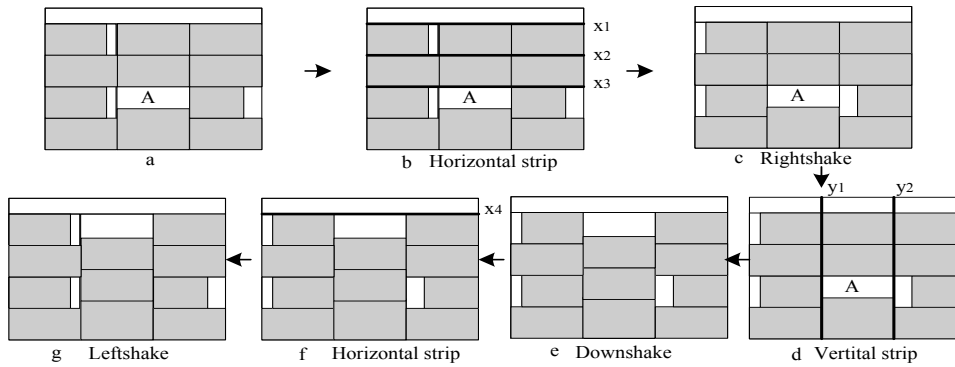


Figure 7 Strip and Shake process

*Shake operator*

Shake the items that have been arranged, making cavities sandwiched between the items move to the top right corner of the stock, and merge with other cavities. According to the packing requirements, each shaking requires three times Shake operator, which are DownShake, Right-Shake and LeftShake. Shaking must be done on the basis of the Strip operator, Horizontal strip needs to be done before LeftShake and Right-Shake, and Vertical Strip needs to be done before DownShake. Shake operator can greatly improve the reuse utilization ratio of the remainders.

For example, as shown in Fig. 7 (The blank parts represent cavities), in order to eliminate the cavity A, firstly, do Horizontal Strip, the initial packing solution will be stripped by the horizontal cutting line  $x_1, x_2, x_3$ , and then RightShake in each strip, the cavities move to the left side of the stock, and then do Vertical Strip, the packing solution will be vertically stripped by  $y_1, y_2$ , DownShake in each strip, at this time, cavity A has been moved to the above of the stock, and merged with the cavities at the top of the stock, because of the place rule (items are placed on the bottom left), after packing, the cavities should be focused on top right of the stock, then again strip, LeftShake in each strip internal, make cavities as far as possible to shift to the right.

**3.3 Second part: Method for avoiding defects**

The algorithm above just discusses the cutting stock problem of stocks without defects. If there are some defects in the stocks, the problem cannot be solved well. This section adds the defect constraint, and extends the algorithm.

For the defects on the stocks, this paper proposes a heuristic algorithm to avoid the defects. In section 3.2.1, the items are given eliminate rule. In section 3.2.2, the boom operator is proposed to deal with the single-defect. Section 3.2.3 introduces a repulsive model to allow the boom operator to be applied to deal with multi-defects.

Finally, the detail procedures to avoid the defects are presented.

**3.3.1 Items eliminate rule: Hit-test**

We do hit-test for the items and the defects.  $D_{lm}$  denotes the  $m$ th defect  $l$ ,  $A_{jk}$  denotes  $k^{th}$  item  $j$ .  $H(D_{lm}, A_{jk})=1$ , if  $dx < w_j + w_l$  and  $dy < h_j + h_l = 0$ ; else.

$H(D_{lm}; A_{jk}) = 1$  means defect  $l$  overlaps with item  $j$ , then the item  $j$  will be eliminated.

**3.3.2 Boom operator**

Generally, the farther away from the defects, the better the material crystal phase structure. Material crystal phase structure affects the quality of the items. In this section, Boom operator is proposed to study this problem.

The aim of boom operator is to improve the utilization ratio of the stock and the quality of the items.

**Definition 6.** Tight Boundary: It is the smallest boundary for the defect, which is formed by four lines. The two horizontal lines nearest to the defect act as the top and bottom boundary, while the two vertical lines nearest to the defect are its left and right boundary. The four lines are sought out by using the cutting line theorem.

Tight boundary and its internal are considered as a whole.

*Process description:* The search process of defect tight boundary is shown in Fig. 8. According to cutting line replacement theorem, this paper explores the segmentation path in the lowest common ancestor from four directions -  $s_1, s_2, s_3$ , and  $s_4$  respectively to find the defect tight boundary which is enveloped by cutting line. From  $s_2$  direction, the first cutting line  $x$  within the scope of the lowest common ancestor is found out. By Continue the search, from  $s_1$  direction, we find  $y_{11}$  (the first replaceable cutting line searched from state  $s_1$ , which replaces  $x$  and becomes the lowest common ancestor's first cutting line). Then the lowest common ancestor

becomes the part of the top right-hand corner composed by  $x$  and  $y_{11}$ , and in turn from  $s_1$  direction,  $y_{12}$  and  $y_{13}$  are found. Repeat the above steps until all the alternative

cutting lines are found and form the tight boundary that can envelope defect.

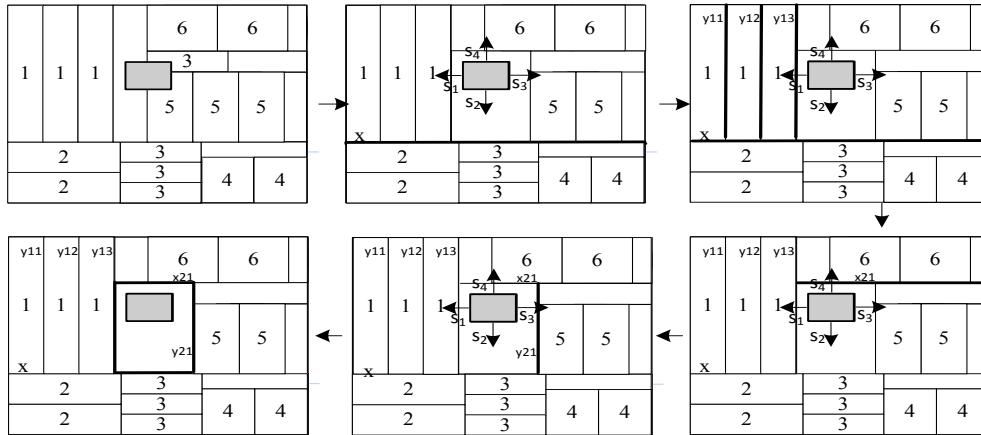


Figure 8 searching process of defect tight boundary

**Outer Boundary:** It is a derivative of tight boundary, which can be obtained by expanding the border of the tight boundary. Outer boundary searching is an inverse process of tight boundary searching, thus Boom in the outer boundary ensures the guillotine cut constraint completely. On the basis of the boundary that has already been found, the outer boundary continually expands to the outside until the outer boundary covers the whole stock.

**Cavities Gathered:** Cavities gathered are under the constraints of the tight boundary and the outer boundary, the action of boom operator and repulsive model, defects will attract the cavities around defects to the nearest place, all the cavities in the boundary will be gathered.

Cavities gather benefits the division and merging of cavities, in this way, here will be more chances to insert the new items.

3.3.3 Repulsive model

The boom operator can only deal with the single-defect, in order to make the algorithm suitable for multi-defects, we adopt repulsive model to extend the algorithm.

Assume that the items of metal and defects of the material are homogeneous, items can be infinite near to the defect, and the defect can be cut. Repulsive  $R_k = g \times \frac{S_{Dk}}{L}$ , where  $g$  instead of repulsive coefficient;  $S_{Dk}$  instead of the area of the  $k^{th}$  defect ( $k= 1;2 \dots ;m$ );  $L$  instead of the distance of the  $k^{th}$  defect center of mass to the items centre of mass.

3.3.4 The procedure of avoiding defect

According to the defect number on the stock, defective stock cutting can be considered in two cases: single-defect stock and multi-defects stock.

(1) Single-defect stock cutting process. The searching process is shown in Fig. 9. The initial state is that the tight boundary of the defect has been searched; the border of the tight boundary is expanded to find the outer boundary; repulsive model is used to attract the cavities going nearest to the defect within the outer boundary; thus one boom operator is completed. Then, take the outer boundary as tight boundary, expand it to find the outer boundary and Boom again like the first time. Repeat the above steps until the whole stock becomes the tight boundary.

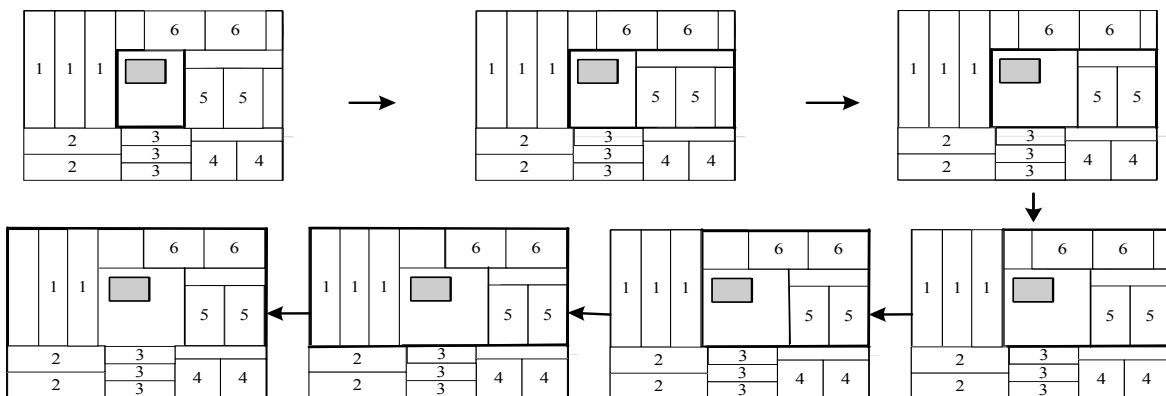


Figure 9 Searching process of single-defect outer boundary

(2) Multi-defects stock cutting process.

If all the defects have the same boundary (tight boundary and outer boundary), the defects can be

regarded as a whole, and this problem can be considered as single-defect stock cutting stock problem.

After eliminating the defects that overlap with items and carrying out the Boom, there will be big cavities

around the defects. According to cutting rules, the items should be continually put into the cavities until the new items cannot be inserted.

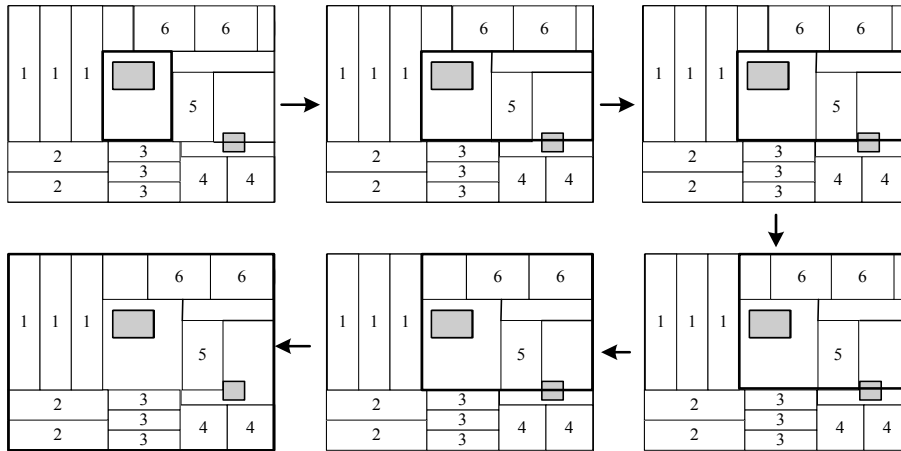


Figure 10 The defects do not have the same boundary

4 Computational experiments

In this section, the heuristic algorithm is tested in two experiments for 2DGCSP and 2DDSGCSP. In Section 4.1, we evaluate the quality of the algorithm for 2DGCSP compared with the algorithm proposed by Vassiliadis (2005) [26]. In Section 4.2, the efficiency of algorithm for 2DDSGCSP is verified through industrial data.

4.1 Experiment for 2DGCSP

For the 2DGCSP, Vassiliadis (2005) [26] gives guillotine cut optimization packing schemes for seven kinds of items (The sizes and quantities of the items are shown in Table 1. Items information table) on a stock, of which the height is 200 and width is unlimited (see Fig. 11). This paper tries to get the initial optimization scheme which ignores the fix orientation of items constraint and the specific packing solution as shown in Fig. 12. All the knife seam width is 2. The required stock sizes, utilization ratio of stock and the quantity of remainders are shown in table 2. Because the reuse of remainders is always decided by the length of its minimum edge, Fig. 13 presents the Frequency Analysis Chart of remainders' minimum edge length.

remainder is supposed to obtain reuse value. From Fig. 13 we can see that in the solution of Vassiliadis' algorithm, only 24,2 % remainders have reuse value. However, in the cutting solution proposed by the paper, the remainders with reuse value rise to 46,2 %.

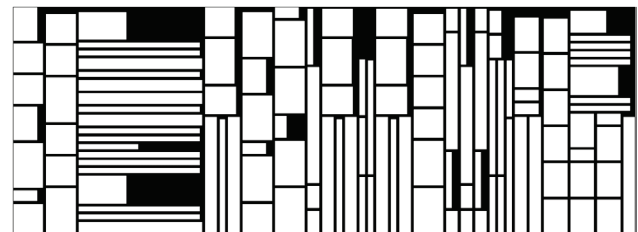


Figure 11 (510x200) Packing result of Vassilios' algorithm

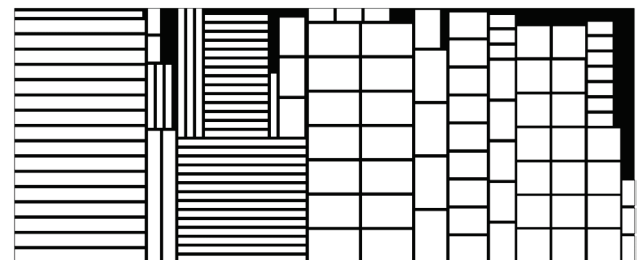


Figure 12 (489x200) Packing result of the proposed algorithm

Table 1 Seven pieces information table

Piece sequence number	Piece specification	Quantity demanded
1	103x10	18
2	101x5	18
3	50x5	18
4	40x25	18
5	30x20	18
6	26x25	18
7	20x10	18

Table 2 Seven pieces guillotine cut result comparison

Sequence	Piece specification	Utilization ratio / %	Quantity Remainers
Fig. 11	510x200	74,74	33
Fig. 12	489x200	77,94	13

From Tab. 2 we can see that the quantity of the remainders has been reduced by 60,6 %. This paper takes 7 (minimum edge 5 adds the knife seam width 2) as benchmark. If the length of the remainder's minimum edge is equal to or greater than the benchmark 7, the

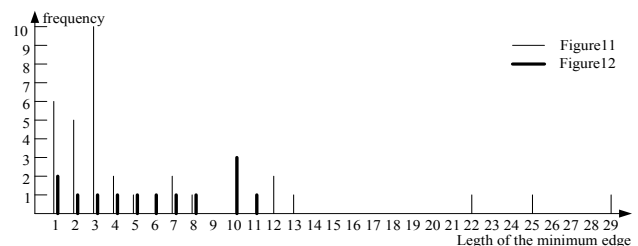


Figure 13 Remainers length of its minimum edge's Frequency Analysis Chart



### 4.2 Industrial data for 2DDSGCSP

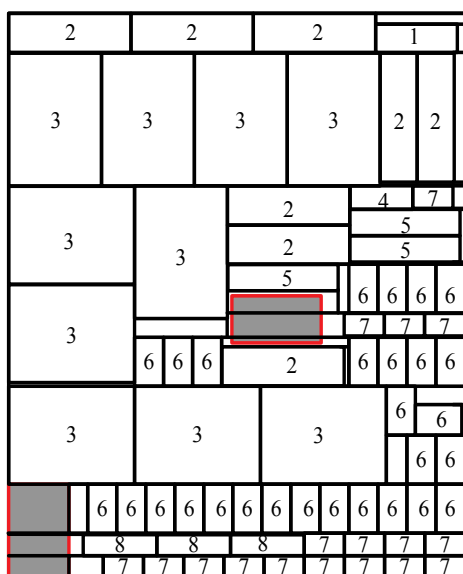
The computational system has been developed on the basis of the algorithm, and has been applied to aircraft manufacturing enterprise, and the practical application effect is very good. In practical application, enterprise packing often involves multi-items multi-sheets, not just a single sheet cutting problem. Tab. 3 is about the basic information of the rectangular items, where are included specifications and quantity, the knife seam width set as 5. Specifications and inventory quantity of the sheets are: 4 pieces of 1022×1200, 1 piece of 1200×1200, 3 pieces of 1200×1500. Tab. 4 is part of the defective-sheets cutting results.

**Table 3** Basic information of the enterprise actual demand pieces

Piece sequence number	Piece specification	Quantity demanded	Piece sequence number	Piece specification	Quantity demanded
1	282×208	20	13	202×100	30
2	280×250	20	14	200×103	15
3	274×189	10	15	183×60	5
4	274×83	30	16	165×45	15
5	267×95	10	17	164×100	20
6	263×153	20	18	150×160	20
7	247×56	15	19	140×50	30
8	235×185	20	20	134×63	5
9	235×69	15	21	120×90	10
10	215×150	10	22	106×84	15
11	210×165	30	23	104×65	30

**Table 4** Enterprise defect plate packing results

Plate specifications	Quantity of plate	Utilization ratio / %
1022×1200	4	92,4
		93,8
		91,2
		92,8
1200×1200	1	93,5
1200×1500	3	91,9
		90,4
		89,7



**Figure 14** Practical example

Fig. 14 shows the cutting results of the proposed algorithm with the stock size 1022×1200 and the two defects (0,0) to (100,135), (500,500) to (700,600). The utilization ratio of stock has reached 92,8 %, from which we can see the computational system developed from the algorithm in practical application is effective, greatly solving the defective-stock cutting stock problem.

From Tab. 4, it can be seen that the practical application of this algorithm is good and the average utilization ratio of the defect stock has reached 91,96 %, solving the enterprise cutting stock problem essentially.

### 5 Conclusion and future research

This paper proposes a new heuristic algorithm to do optimization cutting and designs a series of operators as division, merge, strip, and shake operator to deal with the first type cavity and the special second type cavity, greatly reducing the number of the remainders and improving the utilization ratio of stock and the reuse ratio of the remainders. Furthermore, a Repulsive Model and Boom operators are designed to deal with the defect stock, making the items avoid the defective area as far as possible, which ensures the quality of items and improves the utilization ratio of stock.

From Fig. 12, it can be seen that the cutting result is neat, thus the number of cuts and the rotation frequency of the stock in the cutting process are greatly reduced, which decreases the worker's workload and the machine load. In Fig. 13, we can see the distribution of the length of the remainders' minimum edge, and obviously, the remainders produced by the algorithm are distributed uniformly, which means more remainders may be available. Although the practical application of the proposed algorithm is effective, it still can be improved in the following aspects: seek a better rotation rule and give a further discussion about the merging rules of general second type cavity.

### Acknowledgements

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper. This work was supported by Major International Joint Research Program of the National Natural Science Foundation of China (71020107027), the National Natural Science Foundation of China (Grant no. 71001075) and Doctoral Fund of Ministry of Education of China (20110181110034).

### 7 Reference

- [1] Gilmore, P. C.; Gomory, R. E. Multistage cutting problems of two and more dimensions. // *Operations Research*. 13, 1(1965), pp. 94-119. DOI: 10.1287/opre.13.1.94
- [2] Lodi, A.; Martello, S.; Vigo, D. Recent advances on two-dimensional bin packing problems. // *Discrete Applied Mathematics*. 123, 1(2002), pp. 379-396. DOI: 10.1016/S0166-218X(01)00347-X
- [3] Christofides, N.; Whitlock, C. Algorithm for two-dimensional cutting problems. // *Operations Research*. 25, 1(1977), pp. 30-44. DOI: 10.1287/opre.25.1.30
- [4] Hifi, M. Exact algorithms for the guillotine strip cutting/packing problem. // *Computers & Operations*

- Research. 25, 11(1998), pp. 925-940. DOI: 10.1016/S0305-0548(98)00008-2
- [5] Martello, S.; Monaci, M.; Vigo, D. An exact approach to the strip-packing problem. // *INFORMS Journal on Computing*. 15, 3(2003), pp. 310-319. DOI: 10.1287/ijoc.15.3.310.16082
- [6] Kenmochi, M.; Imamichi, T.; Nonobe, K.; Yagiura, M. Exact algorithms for the two-dimensional strip packing problem with and without rotations. // *European Journal of Operational Research*. 198, 1(2009), pp. 73-83. DOI: 10.1016/j.ejor.2008.08.020
- [7] Macedo, R.; Alves, C.; de Carvalho, J. M. V. Arc-flow model for the two-dimensional guillotine cutting stock problem. // *Computers & Operations Research*. 37, 6(2010), pp. 991-1001. DOI: 10.1016/j.cor.2009.08.005
- [8] Dolatabadi, M.; Lodi, A.; Monaci, M. Exact algorithms for the two-dimensional guillotine knapsack. // *Computers & Operations Research*. 39, 1(2012), pp. 48-53. DOI: 10.1016/j.cor.2010.12.018
- [9] Pisinger, D.; Sigurd, M. The two-dimensional bin packing problem with variable bin sizes and costs. // *Discrete Optimization*. 2, 2(2005), pp. 154-167. DOI: 10.1016/j.disopt.2005.01.002
- [10] Cintra, G. F.; Miyazawa, F. K.; Wakabayashi, Y.; Xavier, E. C. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. // *European Journal of Operational Research*. 191, 1(2008), pp. 61-85. DOI: 10.1016/j.ejor.2007.08.007
- [11] Silva, E.; Alvelos, F.; de Carvalho, J. M. V. An integer programming model for two- and three-stage two-dimensional cutting stock problems. // *European Journal of Operational Research*. 205, 3(2010), pp. 699-708. DOI: 10.1016/j.ejor.2010.01.039
- [12] Castro, P. M.; Oliveira, J. F. Scheduling inspired models for two-dimensional packing problems. // *European Journal of Operational Research*. 215, 1(2011), pp. 45-56. DOI: 10.1016/j.ejor.2011.06.001
- [13] Hopper, E.; Turton, B. C. H. An empirical investigation of meta-heuristic and heuristic algorithm for a 2d packing problem. // *European Journal of Operational Research*. 128, 1 (2001), pp. 34-57. DOI: 10.1016/S0377-2217(99)00357-4
- [14] Burke, E.; Graham, K.; Glenn, W. A new placement heuristic for the orthogonal stock-cutting problem. // *Operations Research*. 52, 4 (2004), pp. 655-671. DOI: 10.1287/opre.1040.0109
- [15] Burke, E.; Graham, K.; Glenn, W. A simulated annealing enhancement of the best-fit heuristic for the orthogonal stock-cutting problem. // *INFORMS Journal on Computing*. 21, 3 (2009), pp. 505-516. DOI: 10.1287/ijoc.1080.0306
- [16] Leung, S. C. H.; Zhang, D.; Sim, K. M. A two-stage intelligent search algorithm for the two-dimensional strip packing problem. // *European Journal of Operational Research*. 215, 1(2011), pp. 57-69. DOI: 10.1016/j.ejor.2011.06.002
- [17] Leung, S. C. H.; Zhang, D.; Zhou, C.; Wu, T. A hybrid simulated annealing meta-heuristic algorithm for the two-dimensional knapsack packing problem. // *Computers & Operations Research*. 39, 1(2012), pp. 64-73. DOI: 10.1016/j.cor.2010.10.022
- [18] Hifi, M.; Negre, S.; Ouafi, R.; Saadi, T. A parallel algorithm for constrained two-staged two-dimensional cutting problems. // *Computers & Industrial Engineering*. 62, 1(2012), pp. 177-189. DOI: 10.1016/j.cie.2011.09.005
- [19] Cui, Y.; Yang, L.; Chen, Q. Heuristic for the rectangular strip packing problem with rotation of items. // *Computers & Operations Research*. 40, 4(2013), pp. 1094-1099. DOI: 10.1016/j.cor.2012.11.020
- [20] Cui, Y.; Yang, L.; Zhao, Z.; Tang, T.; Yin, M. Sequential grouping heuristic for the two-dimensional cutting stock problem with pattern reduction. // *International Journal of production economics*. 144, 2(2013), pp. 432-439. DOI: 10.1016/j.ijpe.2013.03.011
- [21] Gilmore, P. C.; Gomory, R. E. The theory and computation of knapsack functions. // *Operations Research*. 14, 6(1966), pp. 1045-1074. DOI: 10.1287/opre.14.6.1045
- [22] Hahn, S. G. On the optimal cutting of defective sheets. // *Operations Research*. 16, 6 (1968), pp. 1100-1114. DOI: 10.1287/opre.16.6.1100
- [23] Scheithauer, G.; Terno, J. Guillotine cutting of defective boards. // *Optimization*. 19, 1 (1988), pp. 111-121. DOI: 10.1080/02331938808843323
- [24] Twisselmann, U. Rectangles avoiding rectangular defects. // *Applied Mathematics Letters*. 12, 6(1999), pp. 135-138. DOI: 10.1016/S0893-9659(99)00092-0
- [25] Neidlein, V.; André, C. G.; Arenales, M. N.; Wäscher, G. Two-dimensional guillotine packing cutting problems with a single defect-an and/or-graph approach. // *Operations Research Proceedings*. 2008, pp. 85-90. DOI: 10.1007/978-3-642-00142-0\_14
- [26] Vassiliadis, V. S. Two-dimensional stock cutting and rectangle packing: Binary tree model representation for local search optimization methods. // *Journal of Food Engineering*. 70, 3(2005), pp. 257-268. DOI: 10.1016/j.jfoodeng.2003.12.008

#### Authors' addresses

##### **Maozhu Jin, Lecturer Dr.**

Sichuan University  
Business School  
No. 24 South Section 1, Yihuan Road, Chengdu, China, 610065  
E-mail: jinmaozhu@scu.edu.cn

##### **Pen Ge, Lecturer Dr.**

Sichuan University  
Business School  
No. 24 South Section 1, Yihuan Road, Chengdu, China, 610065  
E-mail: gepen@scu.edu.cn

##### **Peiyu Ren, Prof. Dr., corresponding author**

Sichuan University, Business School  
No. 24 South Section 1, Yihuan Road, Chengdu, China, 610065  
E-mail: renpy.scu@163.com