mathematics

MDPI

Article

Optimization Method for Guillotine Packing of Rectangular Items within an Irregular and Defective Slate

Kaizhi Chen [1,2,*], Jiahao Zhuang [1,2], Shangping Zhong [1,2] and Song Zheng [3,4]

1 Department of Mathematics and Computer Science, Fuzhou University, Fuzhou 350100, China; foreverzhuang@163.com (J.Z.); spzhong@fzu.edu.cn (S.Z.)
2 Network System Information Security Fujian Provincial University Key Laboratory, Fuzhou University, Fuzhou 350108, China
3 College of Electrical Engineering and Automation Fuzhou University, Fuzhou University, Fuzhou 350108, China; s.zheng@fzu.edu.cn
4 Key Laboratory of Industrial Automation Control Technology and Information Processing, Fuzhou University, Fuzhou 350108, China
* Correspondence: ckz@fzu.edu.cn

Received: 29 September 2020; Accepted: 28 October 2020; Published: 1 November 2020

Abstract: Research on the rectangle packing problems has mainly focused on rectangular raw material sheets without defects, while natural slate has irregular and defective characteristics, and the existing packing method adopts manual packing, which wastes material and is inefficient. In this work, we propose an effective packing optimization method for nature slate; to the best of our knowledge, this is the first attempt to solve the guillotine packing problem of rectangular items in a single irregular and defective slate. This method is modeled by the permutation model, uses the horizontal level (HL) heuristic proposed in this paper to obtain feasible solutions, and then applies the genetic algorithm to optimize the quality of solutions further. The HL heuristic is constructed on the basis of computational geometry and level packing. This heuristic aims to divide the irregular plate into multiple subplates horizontally, calculates the movable positions of the rectangle in the subplates, determines whether or not the rectangle can be packed in the movable positions through computational geometry, and fills the scraps appropriately. Theoretical analysis confirms that the rectangles obtained through the HL heuristic are inside the plate and do not overlap with the defects. In addition, the packed rectangles do not overlap each other and satisfy the guillotine constraint. Accordingly, the packing problem can be solved. Experiments on irregular slates with defects show that the slate utilization through our method is between 89% and 95%. This result is better than manual packing and can satisfy actual production requirements.

Keywords: defective irregular slate; rectangle packing problems; guillotine cuts; horizontal level packing; computational geometry; genetic algorithm

1. Introduction

In various industries, such as stone, glass, paper, and wood, cutting large pieces of raw materials into rectangular items with different quantities and sizes exhibits practical application contexts. In the stone industry, a large stone plate is directly cut from natural rock blocks, typically with irregular contour and black spot cracks, thereby causing difficulties in stone processing. Natural stone currently uses traditional manual packing, which is time-consuming and exhibits a low utilization rate. The data provided by stone enterprises indicate that the slate utilization of manual packing for irregular and defective stone slates is below 85%. In general, numerous defects and edges of a slate indicate low

efficiency. Therefore, the intelligent packing of irregular and defective slates has become an urgent requirement of the stone industry.

The two-dimensional packing problem is also known as the two-dimensional cutting problem in the stone and other industries. Rectangle packing problem [1], a subproblem of the two-dimensional packing problem, refers to the positioning of several axis-aligned rectangles inside a given plane region to satisfy the necessary constraints (such as the guillotine cutting pattern, in which items must be obtained through a sequence of edge-to-edge cuts) while maximizing its utilization. The rectangle packing problem is a classic NP-hard [2] problem, and the optimal solution cannot be obtained in polynomial time at present. A common solution to this type of problem is using an optimization method for determining a favorable or feasible solution in an acceptable time.

The irregular shape and surface defects of a slate must be considered in the rectangle packing problem in the stone industry. The rectangle can be rotated by 90° (horizontal or vertical), and the cutting pattern is guillotine. At present, the research results on the rectangle packing problem focus on rectangular raw material sheets. Many mathematical models and algorithms have been proposed to solve the rectangle packing problem of rectangular sheets under various constraints. However, to the best of our knowledge, no relevant literature on studying the guillotine packing problem of rectangular items in irregular and defective slates is available. Therefore, our work has practical application significance and provides an effective and feasible solution to the packing problem of irregular slates with defects.

The contributions of our work are presented as follows:

- We first propose the horizontal level (HL) heuristic based on level packing and computational geometry for a single irregular and defective plate to obtain feasible packing solutions. The level packing strategy [3] has been proposed to solve the guillotine packing problem of rectangular plates. The core idea is that the rectangles are packed in rows, which form levels within rectangular plates. The HL heuristic applies the level packing strategy to the packing problem of irregular plates with defects, divides irregular plates into multiple subplates with horizontal lines, and separately packs the subplates through computational geometry to obtain a packing solution that satisfies the constraints;
- Theoretical analysis yields the theorem and confirms that the rectangles obtained through the HL heuristic are inside the plate and do not overlap with the defects; the packed rectangles do not overlap each other and satisfy the guillotine constraint;
- We propose a packing algorithm that combines HL heuristic and genetic algorithm [4] based on the permutation model, and use it as the packing optimization method for natural slate. The packing algorithm optimize the packing solution through continuous iterative search;
- We use the order data provided by stone enterprises to conduct experiments, and evaluate and analyze the performance of the packing algorithm, and confirm that the implementation of our Algorithm effectively satisfies the Theorems in this work.

The remainder of this paper is organized as follows: Section 2 reviews some related works. Section 3 describes the problem and proposes our method. In Section 4, we analyze the algorithm of this work and obtain the statement and verify it. Section 5 evaluates the algorithm through an experiment. Section 6 presents the conclusions drawn from this work.

2. Related Work

2.1. Rectangle Packing Problem

The rectangle packing problem is a typical combinatorial optimization problem. The three main algorithms commonly used to solve the rectangle packing problem are exact algorithm, heuristic, and metaheuristic.

2.1.1. Exact Algorithm

Usually, the mathematical programming model is established based on practical problems, and then the optimal solution is obtained by solving the model. For common mathematical programming models, such as integer linear programming model and mixed integer programming model, the branch and bound algorithm is generally used to solve it. Sarin et al. [5] proposed a hybrid programming model to solve the problem of packing non-identical rectangles in rectangular regions. Beasley [6] abstracted the rectangle packing problem into a 0–1 integer programming problem and solved it. Martello et al. [7] proposed an enumeration method to search the optimal solution of the Two-Dimensional Strip Packing Problem (2SPP). The branching scheme adopted is an improvement on the branch and bound algorithm proposed by Scheithauer [8] and Martello et al. [9]. The computation time and space required by the exact algorithm will exponentially increase with the size of the packing problem.

2.1.2. Heuristic Algorithm

Heuristic is used to seek an intuitive, empirical rule for finding a favorable and feasible solution. The HRBB algorithm proposed by Cui et al. [10] is a heuristic recursive algorithm that combines recursive structure and branch and bound techniques to solve the guillotine packing problem for rectangular strips. Some classic heuristics based on level packing can be used to solve the guillotine packing problem. Examples of such heuristics include the next-fit decreasing height (NFDH) algorithm proposed by Coffman [11]; that is, the item is packed left and justified on the current level (initial bottom) of the rectangular sheet if it fits. Otherwise, a new level is created (as a horizontal line drawn on the top of the tallest item packed on the current level) and then packed. Lodi et al. [12,13] also adopted the level packing strategy and proposed floor–ceiling and knapsack packing algorithms to solve the guillotine packing problem of rectangular boards. Previous studies have shown that the level packing strategy is only used on rectangular plates. Thus, no knowledge of computational geometry is involved. We use the level packing strategy combined with computational geometry to solve the packing problem of irregular and defective plates. Heuristic plays an important role in the actual production packing because it can easily integrate various constraints and specific targets. However, the shortcomings of such heuristics are also evident, and obtaining an excellent packing scheme is typically difficult.

2.1.3. Metaheuristic Algorithm

The commonly used metaheuristics for packing problems are genetic algorithm, simulated annealing [14], tabu search [15], etc. Lodi et al. [16] developed an effective tabu search algorithm for rectangle packing and for variants of the problem involving the possibility of rotating the item by 90° or the additional constraint. Jakobs [17] used the bottom left heuristic as the decoding process of the genetic algorithm and solved the problem of packing rectangular pieces in a rectangular area with fixed length and width. For an orthogonal packing problem, Soke et al. [18] combined the genetic algorithm and the simulated annealing algorithm to search for the optimal rectangle packing sequence. Bortfeldt [19] proposed the SPGAL algorithm based on genetic algorithm to solve the rectangle packing problem of two-dimensional rectangular strips. A favorable solution by exact algorithms or heuristics for the NP-hard optimization problem is difficult to obtain quickly. By contrast, metaheuristics exhibit favorable performance in this respect.

2.2. Rectangle Packing of Irregular Plates

The abovementioned algorithms are all presented for packing on rectangular regions. Few studies and achievements on rectangle packing in irregular or defective regions are available. Research on rectangle packing of irregular plates: Birgin et al. [20] used nonlinear optimization on orthogonal packing of identical rectangles within the arbitrary convex regions. Cassioli et al. [21] proposed a heuristic method based on an iterative local search for the problem of packing the maximum number of identical rectangles within a convex region. An FSS algorithm was proposed by Beasley et al. [22] to

solve the packing problem of rectangle in circular container. Scheithauer et al. [23] solved the problem of guillotine cutting rectangle from rectangular and defective boards with dynamic programming. Jin et al. [24] proposed a heuristic for solving the packing on defective rectangular plates. FSS algorithm was proposed by Beasley et al.

Genetic algorithm is a random global search and optimization method and has been effectively applied to solve various combinatorial optimization problems. A population consists of a set of chromosomes. Each chromosome is a possible solution, and the fitness function measures the quality of each possible solution. Genetic algorithm searches new and better solutions to a problem by improving the current population. In the population update process, the genetic algorithm continuously extracts good genes to obtain a better solution.

2.3. Computational Geometry

Rectangle packing considered both irregular and defective region always encounters the algorithm of Computational Geometry. The following computational geometry problems are encountered in this work, such as how to determine the rectangle is inside the polygon and the rectangle overlaps the polygon. These problems can be decomposed into the following sub-problems to computer, included whether the point is inside the polygon, whether the line segment is inside the polygon, and whether the two polygons overlap. Two common algorithms, namely, winding number [25] and ray casting, are used to determine whether a point within a polygon is called a point in polygon (PIP) problem in computational geometry. Methods to determine whether two polygons overlap are as follows: (a) Calculate intersection area. (b) Calculate no-fit polygon (NFP) [26]. (c) Use the separating axis theorem (SAT) [27].

3. Formulation and Algorithm

3.1. Problem Description

The following notations are the specific symbols used in this work.

Ω:	An irregular plate
Γ:	Set of polygons
B:	The polygon of irregular plate boundaries, represented by a set of polygon clockwise vertex coordinates
B_j:	Divide B horizontally into several subplates; B_j is a subplate numbered j
D:	The set of polygons of irregular plate defects; D_t is the defect numbered t, where $0 \le t \le \omega$
D_t:	The polygon of irregular plate defect, represented by a set of polygon counterclockwise vertex coordinates
R:	The set of rectangles. $R = \{1, \ldots, N\}$, where the element in the set is the serial number of the rectangle, and the negative number represents the rotation of the rectangle by 90°
R_i:	R_i represents a rectangle numbered $\|i\|$
$R_i(x,y)$:	$R_i(x,y)$ indicates that the rectangle R_i is in coordinate (x,y), where x and y correspond to the abscissa and ordinate of the upper left vertex of the rectangle (horizontal axis rightward, vertical axis downward)
h_i:	The vertical height of R_i
w_i:	The horizontal length of R_i

Given an irregular plate Ω, the boundary polygon is B, and the defect polygon set is D. The set of rectangles is R, and the axis-aligned rectangle R_i can be rotated by 90° for packing. The total area of the plate is S_B, and the total area of the defects is S_D. The set of the packed rectangle is K, where $K \subseteq R$.

The constraints are presented as follows:

(I) The packed rectangles cannot overlap with each other;

(II) The packed rectangles must be inside the irregular plate and cannot overlap with the defects;

(III) The packed rectangles must satisfy the guillotine cut constraint.

The rectangle packing problem of the objective function is

$$\max\left\{\sum_{\lambda\in K} h_\lambda \times w_\lambda / (S_B - S_D)\right\}, \tag{1}$$

3.2. Model

Some heuristics construct a feasible pattern for NP-hard packing problems based on a given sequence of the rectangles. Scheithauer [28] mentioned a permutation model that can be used to optimize these problems. In this model, a sequence of rectangles must be determined to ensure that the results obtained using the selected heuristic are optimal. We use this model for optimization.

Given an irregular and defective plate Ω, a rectangle set $R = \{1, \ldots, N\}$. The permutation model for our problem is demonstrated as follows: Find a permutation π^* of R such that the HL heuristic computes a pattern of maximum plate utilization. HL heuristic is based on the rectangle order to obtain a feasible solution for the packing. Let π be a permutation of rectangle sets and HL heuristic output for plate utilization. The optimization target function of the permutation model can be expressed as follows

$$\max HL(\pi, B, D), \tag{2}$$

We use the genetic algorithm to optimize the objective function. GA function is the optimizer. Accordingly, we can obtain the approximate optimal packing solution π^*.

3.3. Algorithm

Definition 1. *Two polygons W and M are provided. If such polygons intersect, then* $W \cap M \neq \varnothing$*; if W is included in M, then* $W \subseteq M$*; if W is separated from M, then* $W \cap M = \varnothing$*; if W and M do not overlap, then* $W \cap M \notin \Gamma$.

Definition 2. *Given a rectangle* R_i *and a subplate* B_j*, the set of movable positions of the rectangle* R_i *at the subplate* B_j *is represented as* $X_i = \{x_{i1}, \ldots, x_{iu}, \ldots, x_{i\rho}\}$*.* X_i *can be stored in a queue, where the head* x_{i1} *is the initial movable position and the rest is the candidate movable position. Due to irregular boundaries and defects, when the position of* R_i *is not available for packing,* R_i *can be moved to the next point to pack. Generally, let* R_k *pack after* R_i*. For the movable positions* X_k *of* R_k *on* B_j*, there is* $x_{k1} \geq x_{i\rho}$.

In Figure 1, the subplate has three shapes, namely, B_1, B_j, and B_C, and exhibits horizontal edges. Here, we take B_1 (the top edge of the polygon is a horizontal line) as an example to calculate the movable positions. The ordinate of the horizontal line of the B_1 is y_1. We align the top edge of R_i with the horizontal line of B_1 and calculate the intersection of B_1 and $y = y_1 + h_i$ as the movable positions (Figure 2). Moreover, we must update X_i by removing elements from the queue that is smaller than the current packing point and adding current packing point to X_i as the initial movable position. If defects are present in the subplate, then the rightmost points of these defect polygons are added to X_i as the candidate moveable point, see Figure 3. When calculating the first packing rectangle of a subplate, the initial movable position of the rectangle must be adjusted in accordance with the subplate shape.

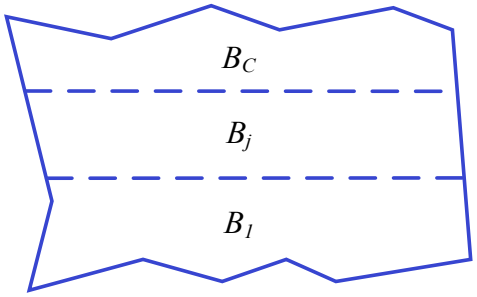

Figure 1. Horizontal split plate.

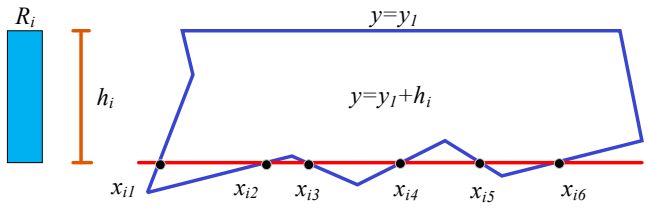

Figure 2. Movable positions of the rectangle in the sub-plate.

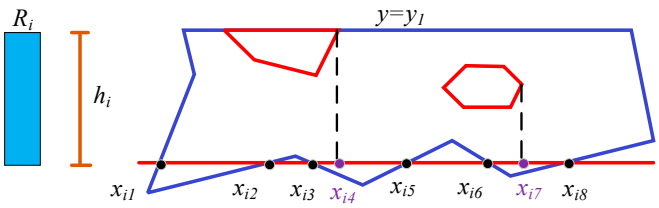

Figure 3. Movable positions of the rectangle in the defective subplate.

Algorithm 1 is the HL heuristic, which divides the irregular plate into multiple subplates with horizontal lines and packs in accordance with the given rectangle order.

Algorithm 1 HL heuristic

Input: boundary polygon B, defect polygon set D, rectangle sequence π
(1) Swap the height and width of the rectangle if the number is negative. Let R' be the next rectangle to be packed and K be the set of packed rectangles.
(2) Find the second lowest vertex of the plate B, and subtract the height h' of R' from the ordinate to obtain a horizontal line, thereby dividing B into the upper subplate B_R and the lower subplate B_P; B_P is the current packing region (see Figure 4).
(3) Repeat (3.1) and (3.2) until B_P finishes its packing.
(3.1) Calculate the movable positions X of R' at B_P.
(3.2) R' moves along the horizontal line of B_P. If $\exists x \in X$, such that R' is within B_P and does not overlap with D, then add R' to K, remove R' from π, and update R'. Otherwise, B_P packing is complete. If the height of the next rectangle R_i is greater than h', then search for R_t whose height is below h'. Exchange the position of R_i and R_t in π, and update $R' = R_t$.
(4) Update $B = B_R$. Repeat (2), (3), and (4) for B_R until B_R does not constitute a polygon.
(5) Find the maximum inscribed rectangles for the waste space around plate boundaries and defects. Then, use the NFDH algorithm to pack the remaining rectangles, and add the packed rectangles to K.
(6) Calculate plate utilization.
$HL(\pi, B, D) = \sum_{\lambda \in K} h_\lambda \times w_\lambda / (S_B - S_D)$

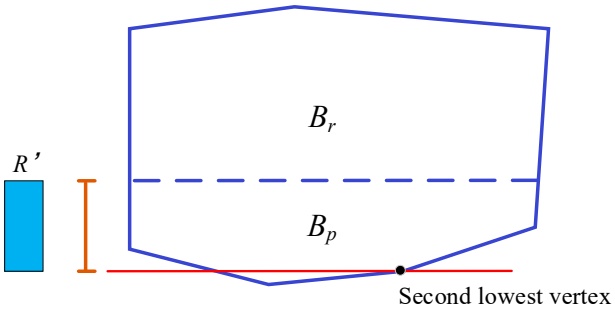

Figure 4. Algorithm 1 step (2): split the plate.

Algorithm 2 is the packing algorithm in this work. Genetic algorithm is utilized as the optimizer, and the rectangle sequence is used as the gene coding. The HL heuristic is adopted as the decoding process.

Algorithm 2 Genetic algorithm + HL heuristic.

Input: boundary polygon B, defect polygon set D, rectangle set $R = \{1, \ldots, N\}$
Initialization: population number M, round iteration number T, crossover probability P_c, mutation probability P_m
(1) Initializing population P (the set of feasible solutions): randomly generate M integer permutations from 1 to N. Each integer randomly generates positive and negative signs. (negative signs mean the rectangle is rotated 90 degrees)
(2) For $\forall \pi \in P$, perform Algorithm 1 to calculate the fitness of π as $f(\pi) = \mathrm{HL}(\pi, B, D)$. Let π^* be the optimal solution of population P.
(3) Repeat (3.1) and (3.2) until the update of new population $\widetilde{P}$ is complete (see Figure 5).
(3.1) Two individuals are selected from P using the method of Roulette Wheel Selection. A new solution is constructed by a two-point crossover with the cross probability of P_c.
(3.2) An individual is randomly selected from P, and the two-point mutation is performed with the mutation probability of P_m.
(4) For $\forall \pi \in \widetilde{P}$, perform Algorithm 1 to calculate the fitness $f(\pi)$ of π.
(5) If a feasible solution $\mu \in \widetilde{P}$ with $f(\pi^*) < f(\mu)$ is obtained, then replace $\pi^* = \mu$. And set $P = \widetilde{P}$.
(6) If the population iteration number reaches T, then stop. The optimal packing is π^*. Otherwise, continue to (3)

In step 3 of Algorithm 2, individuals must be a permutation with unduplicated numbers from R. Conventional bitwise crossover does not guarantee that permutations are unduplicated. Therefore, we use a scan operator to fill each number in the new sequence after crossover operator. We use an example to describe the specific steps of crossover and mutation operator as Figure 5.

Step 1: Two individuals {5, −1, 3, −4, 7, 6, −2}, {3, 4, 2, −5, 1, −7, 6}, Where N = 7, exchange the middle 3 number {3, −4, 7} and {2, −5, 1} to each son sequence;

Step 2: Then, scan each parent sequence to put the reminder number into the son sequence. As Parent 1, scan the middle number {3, −4, 7} and other reminder number {5, −1, 6, −2} in turn. This process must ensure the numbers filled in are not identical to the son middle numbers {2, −5, 1}, so the reminder number sequence {3, −4, 7, 6} are filled in Son 1. New individuals are generated;

Step 3: Each new individual carries out the mutation operator, as in Figure 5b.

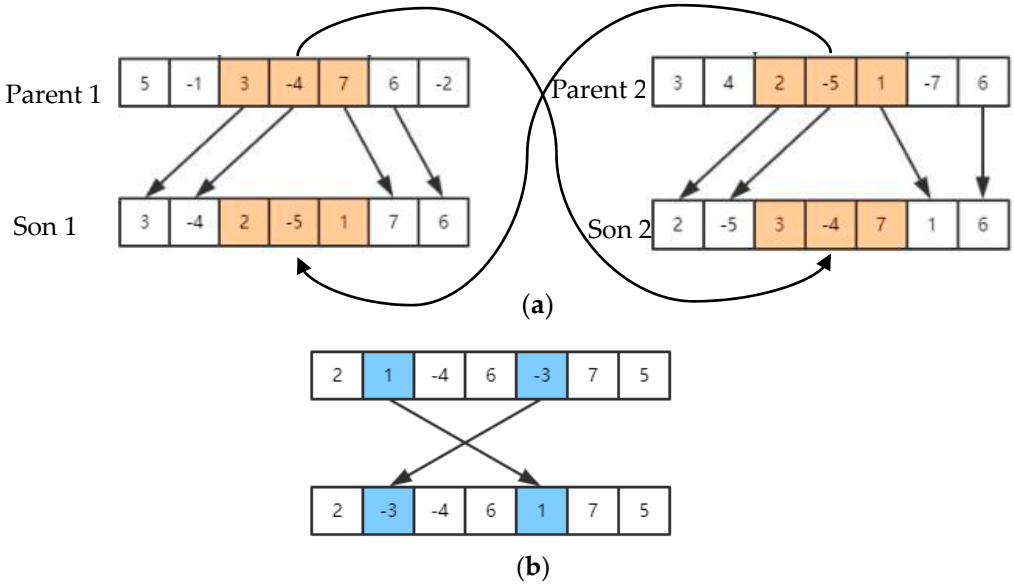

Figure 5. Algorithm 2 step (3): (**a**) tow-point crossover operator. (**b**) mutation operator.

If population P has 200 individuals, it can be divided into 100 groups. Each group with two individuals can carry out one crossover operator, and each individual can carry out one mutation operator. Therefore, the total times that the actions (3.1) and (3.2) are performed are 100 and 200, respectively.

4. Algorithm Analysis

The genetic algorithm, an optimizer of the permutation model, has favorable global search ability. Such an algorithm maps the rectangle sequence to the encoding of the solution. Thus, the derivative and function continuity have no limitation.

The HL heuristic has two basic ideas. First, a horizontal line must be determined, and the irregular plate must be divided into two parts. For the lower part, pack the selected rectangle along the horizontal line. The rectangle must be moved to the next movable position when encountering defects or irregular edges. These steps are repeated for the remaining parts of a plate. Second, waste is inevitable after the initial pack, given the existence of irregular boundaries and defects. We must use this part of the plate as much as possible. The main method is to obtain the maximum inscribed rectangle of the waste material after defect removal and use the NFDH algorithm proposed by Coffman [11] for a simple fill.

Let $\widetilde{P}$ be the population after the iterative T' round of genetic algorithm. Let π be a permutation of R and $\pi \in \widetilde{P}$. For the irregular plate Ω (Figure 6), the boundary polygon B and the set of defect polygons D are represented by blue and red, respectively. The boundary polygon is divided into C subplates, that is, $B = \{B_1, \ldots, B_j, \ldots, B_C\}$. Let $U = \{U_1, \ldots, U_j, \ldots, U_C\}$, U_j be the set of rectangles to be packed corresponding to the subplate B_j, and $U_j = \pi$, where $U_1 = \pi$, that is, B_1 is the initial layout subplate.

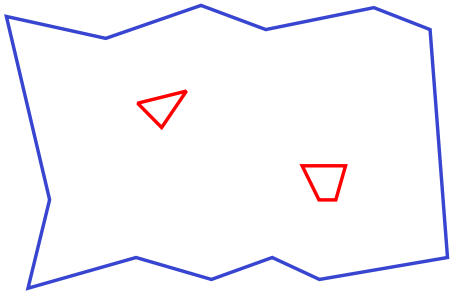

Figure 6. Irregular and defective plate.

Function F (R_i, B_j, x, y) is used to determine whether R_i is inside B_j at coordinate (x, y). Function Q (R_i, D_t, x, y) is used to determine whether R_i overlaps with defect polygon D_t at coordinate (x, y), where $0 \le t \le \omega$. The specific definition is provided as follows

$$F(R_i, B_j, x, y) = \left\{ \begin{array}{l} 1, R_i(x, y) \subseteq B_j \\ 0, otherwise \end{array} \right\}, \tag{3}$$

$$Q(R_i, D_t, x, y) = \left\{ \begin{array}{l} 1, R_i(x, y) \cap D_t \notin \Gamma \\ 0, otherwise \end{array} \right\}, \tag{4}$$

R_i moves along the horizontal line y_j of B_j. Function G (R_i, B_j, D, X_i) is used to determine whether R_i can be packed in B_j. The first packing area of B_j is denoted as S_j (U_j, B_j, D). Then,

$$G(R_i, B_j, D, X_i) = \left\{ \begin{array}{l} 1, \sum\limits_{x \in X_i} \prod\limits_{t=0}^{\omega} F(R_i, B_j, x, y_j) \times Q(R_i, D_t, x, y_j) \\ 0, otherwise \end{array} \right\} \tag{5}$$

$$S_j(U_j, B_j, D) = \sum_{i \in U_j} h_i \times w_i \times G(R_i, B_j, D, X_i), \tag{6}$$

Let I be the set of rectangles successfully packed for the first time, and $\pi - I$ be the set of remaining rectangles. Let Z be rectangular wastes without defects, and the NFDH algorithm is used to obtain the packing area of the waste material. We can obtain the total packing area as follows

$$S(\pi, B, D) = \sum_{j=1}^{C} S_j(U_j, B_j, D) + NFDH(\pi - I, Z), \tag{7}$$

We pack rectangles π on irregular plates Ω through HL heuristic and obtain the plate utilization as follows

$$Rate = HL(\pi, B, D) = \frac{S(\pi, B, D)}{S_B - S_D}, \tag{8}$$

We can derive the following two Theorems through the abovementioned analysis.

Theorem 1. *The rectangles obtained through the HL heuristic are inside the plate and do not overlap with the defects.*

Prove. Let K be the set of rectangles obtained through the HL heuristic. Equation (7) denotes that K is composed of $K - I$ and I. $K - I$ satisfies Constraint (II) in Section 3 because it is obtained through the NFDH algorithm. Let $\lambda \in I$. Equations (5) and (6) express that $R_\lambda \subseteq B_j$, and $B_j \subseteq B$; thus, $R_\lambda \subseteq B$, that is, R_λ is inside B. Given that $\forall\ D_t \in D$, and $R_\lambda \cap D_t \notin \Gamma$, R_λ does not overlap with the defects. Then, $\forall\ \lambda \in I$, and $(R_\lambda \subseteq B) \wedge (R_\lambda \cap D_t \notin \Gamma)$; thus, I satisfies Constraint (II) in Section 3. Therefore, rectangles in K are inside the plate and do not overlap with the defects. □

Theorem 2. *The rectangles obtained through the HL heuristic do not overlap each other and satisfy guillotine constraint.*

Prove. Let K be the set of rectangles obtained by HL heuristic. K is composed of $K - I$ and I. $K - I$ satisfies Constraints (I) and (III) in Section 3 because it is obtained through the NFDH algorithm. The rectangles in I are packed along the horizontal line of subplate; thus, the pack conforms to a guillotine pattern. Accordingly, Constraint (III) in Section 3 is satisfied. Let R_i (x_i, y) and R_t (x_t, y) be two rectangles of the same level in I, and R_t (x_t, y) is packed after R_i (x_i, y). The movable position indicates that $x_t \ge x_i$. Therefore, the rectangles in I do not overlap each other. Thus, Constraint (I)

in Section 3 is satisfied. Consequently, the rectangles in K do not overlap each other and satisfy the guillotine constraint. □

In this section, we demonstrate that the HL heuristic can be used to obtain a layout scheme that satisfies the constraints through theoretical analysis. Given that the Algorithm 2 uses the HL heuristic as the decoding process, its packing result must satisfy Theorems 1 and 2. Thus, the packing result is accurate. In the next section, we will further verify the effectiveness of the Algorithm 2 and evaluate its performance through experiments.

5. Experiments

In this section, we will use the Algorithm 2 to conduct experiments through the actual data provided by stone enterprises, and analyze the algorithm according to the experimental results.

5.1. Dataset and Experimental Environment

The experimental data in this work are mainly provided by stone enterprises. They are composed of rectangular stone orders and original stone pictures. Rectangular stone orders contain a total of 272 rectangles in 49 specifications. Table 1 summarizes the partial rectangle information. Each rectangle will be renumbered, with numbers ranging from 1 to 272. Another program that we implemented can obtain the vertex coordinates of the slate through the slate image. Six slates, which are approximately 3×3 m in size, are utilized in this work. Part of the slate image is illustrated in Figure 7. The data can be downloaded from https://github.com/chkz/Plate-data.

The experiments were run on a platform with 3.6 GHz Intel Core i7 processor, 8 GB of RAM, and Windows operating system. The development language is C++.

Table 1. Partial rectangle information.

Type	Width (mm)	Height (mm)	Quantity
4	1050	477	4
5	950	477	4
6	950	244	2
7	850	244	6
8	197	897	2
9	193	897	8

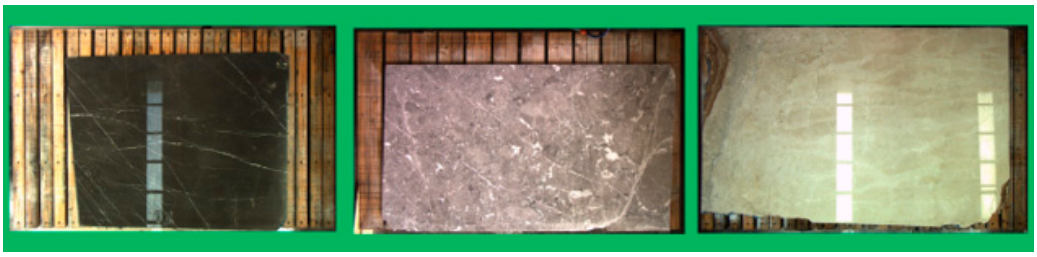

Figure 7. Partial slate image.

5.2. Parameters Experiments Analysis

There are several important parameters in our algorithm: population number M, round iteration number T, crossover probability P_c, mutation probability P_m. We use experiments to determine the optimal parameters.

To find the optimal values by the traversal method, we set round iteration number T = [10 20 30 40 50 100 150 200 250 300], population number M = [50 100 150 200 250 300], crossover probability P_c = [0.2 0.4 0.6 0.8], mutation probability P_m = [0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08].

First, test the relationship between the round iteration number and performance of plate utilization rate. Part of the results are as shown in Figure 8. Generally, the more iterative rounds there are, the better the performance in all results, but under the current samples, the utilization rate tends to be stable after almost 100 rounds, with only a small improvement, and even some jitters. Therefore, we set the round iteration number of our algorithm to 100 in the subsequent experiments.

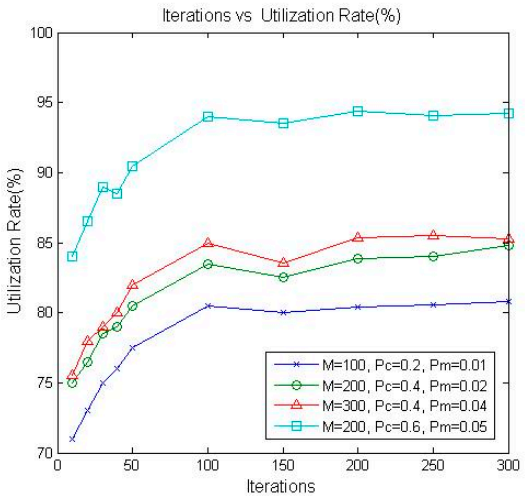

Figure 8. Relationship between iterations and utilization rate.

The relationship between the population number M and performance is as shown in Figure 9. We set the round iteration number to be fixed at 100. Among the three lines in the figure, two lines are the slates' utilization rate under the randomly selected fixed parameters P_c and P_m, and the other is the average value of the slates' utilization rate of all P_c and P_m under a different M. It can be seen from the figure that the performance of the population number is better when it is 200−250, hence, we chose $M = 200$ for our algorithm in the subsequent experiments.

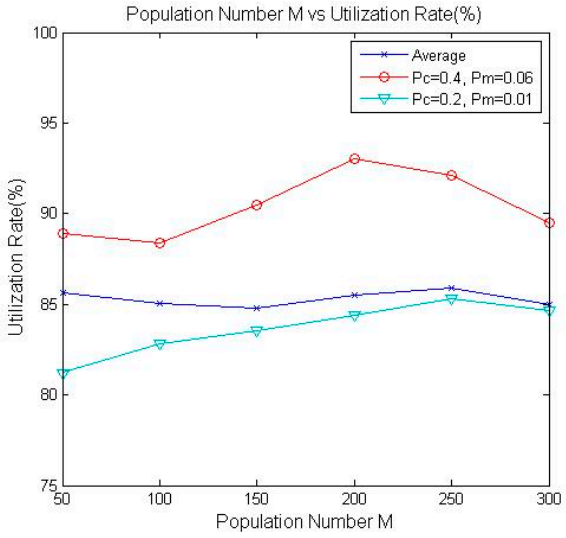

Figure 9. Relationship between population number and utilization rate.

For the last two parameters, we traverse P_c and P_m under the condition of $T = 100$, $M = 200$. It can be seen from the Figure 10 that when $P_c = 0.6$, $P_m = 0.05$, the slates utilization rate is the best. Therefore, these optimal parameter value are used in all the subsequent experiments.

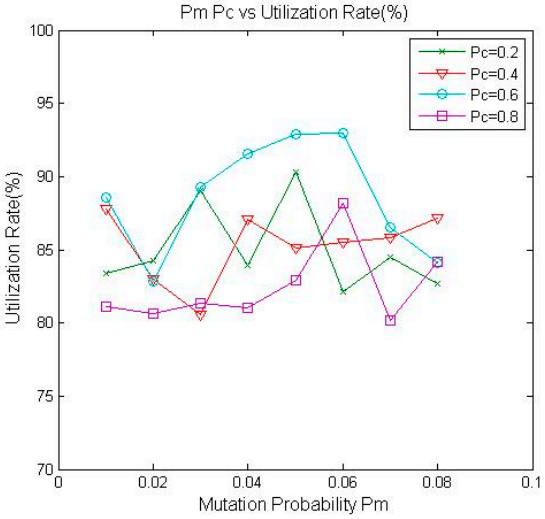

Figure 10. Relationship between P_c, P_m and utilization rate.

5.3. Packing Results

In Figures 11–13, the black polygon is the plate, the red polygon is the defect, and the blue fill is the packed rectangle. The top left corner of the pictures represent the plate utilization, and the axis is in mm.

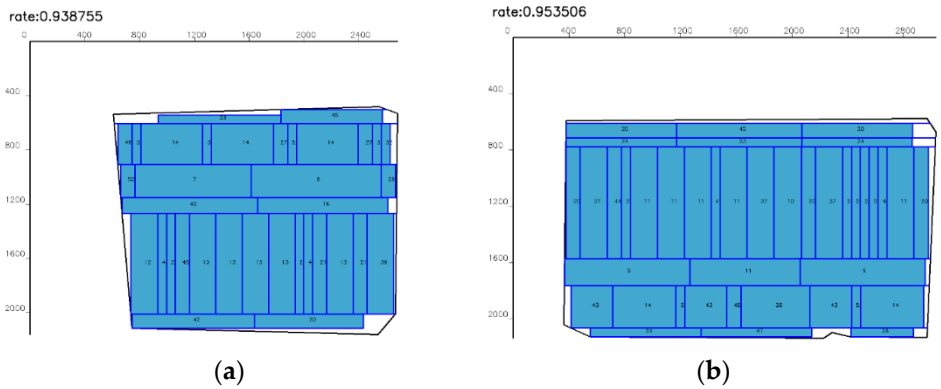

Figure 11. Packing results of non-defective irregular plates. (**a**) Plate A result. (**b**) Plate B result.

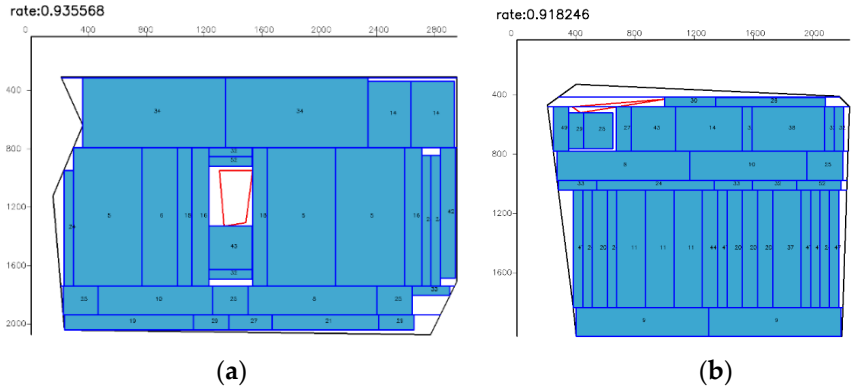

Figure 12. Packing results of irregular plates with a defect. (**a**) Plate C result. (**b**) Plate D result.

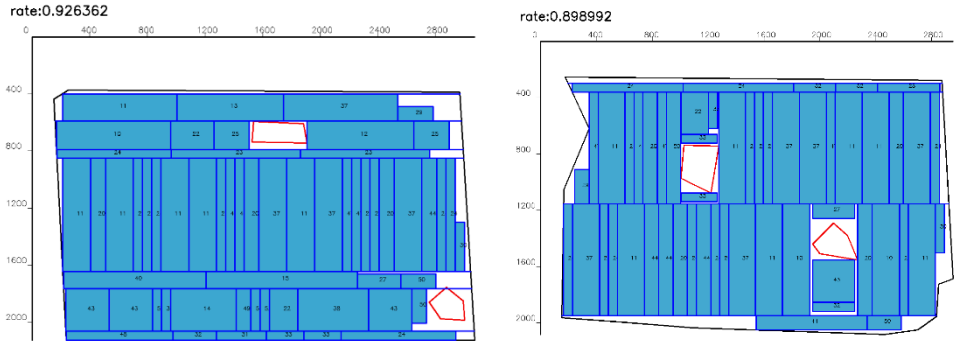

Figure 13. Packing results of irregular plates with two defects. (**a**) Plate E result. (**b**) Plate F result.

From Figures 11–13, we can see that the packing results conform to the two previous constraints. All rectangles are inside the panel and avoid surface defects, consistent with Theorem 1 in Section 4. The two adjacent rectangles do not overlap each other, and all rectangles are packed in levels, consistent with Theorem 2 in Section 4.

In summary, the experimental results verify the effectiveness of the algorithm and the implementation is consistent with Theorems 1 and 2. Therefore, Algorithm 2 can satisfy the constraints for different irregular plates with defects and obtain an effective layout scheme.

5.4. Comparative Analysis

To our best knowledge, no relevant literature on the guillotine packing of rectangles in irregular and defective plates is available at present. In this work, we propose the packing algorithm for the single nature slate. Therefore, we compare Algorithm 2 with the algorithms of other packing problems, as well as the manual packing of a stone enterprise.

We have selected some relatively new packing algorithms that give complete experimental results for comparison. At the same time, the mean waste rate and mean run time of different methods are calculated and used as the main performance indicators. Table 2 gives a comparison of the different packing methods. SPGAL [19] is based on genetic algorithm; HRBB [10] algorithm is a heuristic recursive algorithm; Jin et al. [24] proposed a heuristic algorithm to solve the packing problem of rectangular plates with defects. In addition, the FSS [22] algorithm and the method proposed by Birgin et al. [20] are exact algorithms.

Table 2. Comparison of different packing methods.

Approach (Author)	Plate Type	with Defects	Pack Type	Mean Waste Rate (%)	Mean Run Time (s)
SPGAL	Rectangular strip	NO	OF	2.41	1.59
HRBB	Rectangular strip	NO	RG	1.2	15.68
Jin et al.	Rectangular stock	YES	RG	6.5	6.2
FSS	Circular container	NO	RF	28.86	43,838
Birgin et al.	Convex region	NO	RF	18.21	3879
Algorithm 2	Irregular slate	YES	RG	7.68	404.25
Manual packing	Irregular slate	YES	RG	>15	Long time

(O: the orientation of the rectangles is fixed, R: the rectangles may be rotated by 90°, F: no guillotine cutting is required, G: guillotine cutting is required).

As can be seen from the above table, SPGAL and HRBB are two algorithms for solving the packing problem of non-defective rectangular plates with different constraints (OF, RG). Both SPGAL and HRBB have a low mean waste rate and mean run time, so good solutions can be obtained quickly. Among them, the mean waste rate of SPGAL is 2.41%, and that of HRBB is 1.2%. For the packing problem of rectangular plates with defects and the packing type is RG, the mean waste rate of the

method proposed by Jin et al. [24] is 6.5%. The mean waste rate increases slightly compared with HRBB, and the quality of the solution decreases a little. Secondly, for circular containers and convex areas, FSS algorithm and the packing method proposed by Birgin et al. [20] have a higher mean waste rate of more than 18%. Since these two algorithms are exact algorithms, their mean run time is much higher than other algorithms, both above 3000 s. In contrast, the mean waste rate of Algorithm 2 for irregular and defective plates is 7.68%, which is slightly higher than the above-mentioned packing methods for rectangular plates. Besides compared with the packing methods for circular containers and convex regions, the mean run time of Algorithm 2 is greatly reduced. This shows that our method can obtain a relatively good solution in the acceptable time for the packing problem of irregular plates with defects.

In addition, the accuracy of manual packing is poor, and the packing efficiency is low. Furthermore, the plate utilization rate is generally below 85%. Our algorithm ensures the effectiveness of packing while significantly improving the plate utilization rate and considerably reducing the time cost of the packing in comparison with manual packing. The plate utilization for general irregular plates can reach more than 90%. By contrast, the plate utilization for irregular and non-defective plates can reach 95%. The packing efficiency of manual packing for plates with numerous irregular boundaries and defects is generally low. The algorithm for these plates can still obtain a good packing scheme in a short time. Therefore, the packing algorithm has a strong practicability.

5.5. Comparison with Jin's Algorithm

The algorithm proposed by Jin et al. [24] is chosen for comparison of performance. The problem it solves has some similarities with our algorithm, such as the same packing type RG, the plates being with defects, and many rectangular specifications in the order. The difference is that both the plate shape and the defect area are rectangular in Jin's problem.

In order to compare comprehensively, we first compare the performance of the two algorithms on the rectangular plates with rectangular defects (Dataset A). Then, we compare the irregular plates with the irregular defects (Dataset B). As shown in Table 3, Dataset A contains two plates with no defects, two plates with one defect, and two plates with two defect areas. The polygonal plates still use the above six Plates (Plate A–F) in Figures 11–13. The length and width of the plates are evenly distributed between 2.5 and 3.0 m. The length and width of the rectangular defect areas are evenly distributed between 10 and 80 cm. These size ranges of the plates and defect are commonly encountered in stone enterprises. The rectangles still adopt the order of 272 rectangles in the 49 specifications introduced above. In order to be compatible with Jin's algorithm in Dataset B, the polygonal plates and the polygonal defect areas must be transformed into rectangles, that is, the largest rectangle is labeled in the polygonal plates by the computational geometry method, and then the smallest rectangle is labeled to cover the polygonal defect areas.

According to the results shown in Table 3, the Algorithm 2 we proposed is better than Jin's algorithm by about 1% in utilization rate on the rectangular plates with rectangular defects, but Jin's algorithm is much faster than our Algorithm 2, as shown in Table 2. The reason for this is that Jin's algorithm is similar to our Algorithm 1, which is a heuristic algorithm. Our Algorithm 2 actually iteratively tried tens of thousands of Algorithm 1 to find the best performance. Therefore, even under the rectangular plates, our Algorithm 2 performs better than Jin's. In addition, the utilization rate of the polygonal plates has increased by 5–7% in Dataset B. The reason for this is that when Jin's algorithm is applied, the polygonal plates must first be cut to the largest rectangle. The edge part cut in this process may be utilized in Algorithm 2, so the utilization rate of Algorithm 2 is improved.

Table 3. Comparison of Jin's Algorithm and our Algorithm 2.

Plate		Defects	Utilization Rate (%)	
			Jin's	Algorithm 2
Dataset A	Plate 1	0	93.3251	94.4525
	Plate 2	0	93.4261	94.6744
	Plate 3	1	92.7524	93.2545
	Plate 4	1	92.6742	93.9825
	Plate 5	2	91.8734	92.2525
	Plate 6	2	90.8543	91.3456
Dataset B	Plate A	0	88.3251	93.8755
	Plate B	0	90.4261	95.3506
	Plate C	1	84.6238	93.5568
	Plate D	1	84.4615	91.8246
	Plate E	2	84.2675	92.6362
	Plate F	2	81.8765	89.8992

6. Conclusions

The stone material increases the complexity of rectangle packing given its physical characteristics. The packing problem in this work is an NP-hard problem and has numerous local extremum points. We propose a packing algorithm that combines HL heuristic and genetic algorithm to solve this problem. The experimental results show that the algorithm is effective and has favorable packing performance. The algorithm can obtain excellent packing results with a relatively short time loss for different irregular slates with defects. Therefore, our approach can reduce the impact of manual factors, such as subjectivity and experience, improve the packing efficiency, and save the use of raw materials, thereby enhancing the economic benefits of the company.

In the actual production of stone enterprises, many slates are frequently used given the variety and quantity of rectangles in the order. Therefore, we will consider the simultaneous packing problem of multiple irregular and defective slates in the future work. We will also look for a packing method that is more consistent with actual production requirements.

Author Contributions: Conceptualization, K.C., J.Z., and S.Z. (Shangping Zhong); methodology, K.C, S.Z. (Shangping Zhong); validation, J.Z.; formal analysis, K.C., J.Z.; investigation, K.C., S.Z. (Shangping Zhong), and S.Z. (Song Zheng); data curation, J.Z.; writing—original draft preparation, J.Z.; writing—review and editing, K.C, J.Z, S.Z. (Shangping Zhong) and S.Z. (Song Zheng); visualization, J.Z.; supervision, K.C., S.Z. (Shangping Zhong). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC), Grant Number 61972187; the Scientific Research Project of Science and Education Park Development Center of Fuzhou University, Jinjiang, Grant Number 2019-JJFDKY-53 and the Tianjin University-Fuzhou University Joint Fund, Grant Number TF2020-6.

Acknowledgments: Thanks to the editors and reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pietrobuoni, E. Two-Dimensional Bin Packing Problem with Guillotine Restrictions. Ph.D. Dissertation, Alma Mater Studiorum Università di Bologna, Bologna, Italy, 2015.
2. Furini, F.; Malaguti, E.; Thomopulos, D. Modeling two-dimensional guillotine cutting problems via integer programming. *INFORMS J. Comput.* **2016**, *28*, 736–751. [CrossRef]
3. Lodi, A.; Martello, S.; Monaci, M. Two-dimensional packing problems: A survey. *Eur. J. Oper. Res.* **2002**, *141*, 241–252. [CrossRef]
4. Kramer, O. *Genetic Algorithm Essentials*; Springer: Cham, Switzerland, 2017; Volume 679.
5. Chen, C.-S.; Sarin, S.; Ram, B. A mixed-integer programming model for a class of assortment problems. *Eur. J. Oper. Res.* **1993**, *65*, 362–367. [CrossRef]

6. Beasley, J.E. An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. *Oper. Res.* **1985**, *33*, 49–64. [CrossRef]
7. Martello, S.; Monaci, M.; Vigo, D. An exact approach to the strip-packing problem. *INFORMS J. Comput.* **2003**. [CrossRef]
8. Scheithauer, G. *Equivalence and Dominance for Problems of Optimal Packing of Rectangles*; Dresden University of Technology: Dresden, Germany, 1997.
9. Martello, S.; Pisinger, D.; Vigo, D. Three-dimensional bin packing problem. *Oper. Res.* **2000**. [CrossRef]
10. Cui, Y.; Yang, Y.; Cheng, X.; Song, P. A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem. *Comput. Oper. Res.* **2008**, *35*, 1281–1291. [CrossRef]
11. Coffman, E.G., Jr.; Garey, M.R.; Johnson, D.S.; Tarjan, R.E. Performance Bounds for Level-Oriented Two-Dimensional Packing Algorithms. *SIAM J. Comput.* **1980**, *9*, 808–826. [CrossRef]
12. Lodi, A.; Martello, S.; Vigo, D. Heuristic and Metaheuristic Approaches for a Class of Two-Dimensional Bin Packing Problems. *INFORMS J. Comput.* **1999**, *11*, 345–357. [CrossRef]
13. Lodi, A.; Martello, S.; Vigo, D. Neighborhood Search Algorithm for the Guillotine Non-Oriented Two-Dimensional Bin Packing Problem. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*; Springer: Boston, MA, USA, 1999; pp. 125–139.
14. Kirkpatrick, S. Optimization by simulated annealing: Quantitative studies. *J. Stat. Phys.* **1984**, *34*, 975–986. [CrossRef]
15. Glover, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **1986**, *13*, 533–549. [CrossRef]
16. Lodi, A.; Martello, S.; Vigo, D. Approximation algorithms for the oriented two-dimensional bin packing problem. *Eur. J. Oper. Res.* **1999**, *112*, 158–166. [CrossRef]
17. Jakobs, S. On genetic algorithms for the packing of polygons. *Eur. J. Oper. Res.* **1996**, *88*, 165–181. [CrossRef]
18. Soke, A.; Bingul, Z. Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems. *Eng. Appl. Artif. Intell.* **2006**, *19*, 557–567. [CrossRef]
19. Bortfeldt, A. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *Eur. J. Oper. Res.* **2006**, *172*, 814–837. [CrossRef]
20. Birgin, E.G.; Martínez, J.M.; Nishihara, F.H.; Ronconi, D.P. Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization. *Comput. Oper. Res.* **2006**, *33*, 3535–3548. [CrossRef]
21. Cassioli, A.; Locatelli, M. A heuristic approach for packing identical rectangles in convex regions. *Comput. Oper. Res.* **2011**, *38*, 1342–1350. [CrossRef]
22. López, C.O.; Beasley, J.E. Packing unequal rectangles and squares in a fixed size circular container using formulation space search. *Comput. Oper. Res.* **2018**, *94*, 106–117. [CrossRef]
23. Scheithauer, G.; Terno, J. Guillotine cutting of defective boards. *Optimization* **1988**, *19*, 111–121. [CrossRef]
24. Jin, M.; Ge, P.; Ren, P. A new heuristic algorithm for two-dimensional defective stock guillotine cutting stock problem with multiple stock sizes. *Teh. Vjesn. Tech. Gaz.* **2015**, *22*, 1107–1116. [CrossRef]
25. Alexander, J.W. Topological Invariants of Knots and Links. *Trans. Am. Math. Soc.* **1928**, *30*, 275. [CrossRef]
26. Adamowicz, M.; Albano, A. Nesting two-dimensional shapes in rectangular modules. *Comput. Des.* **1976**, *8*, 27–33. [CrossRef]
27. Scholtes, S. Nondifferentiable and two-level mathematical programming. *Eur. J. Oper. Res.* **1997**, *102*, 244–245. [CrossRef]
28. Scheithauer, G. *Introduction to Cutting and Packing Optimization*; Springer: Cham, Switzerland, 2018; Volume 263.