Research Article

Descending Packing Algorithm for Irregular Graphics Based on Geometric Feature Points

Pengfei Zheng ^{1,2}, Jingjing Lou ², Dajun Lin,¹ and Qi An ¹

¹*School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai 200237, China*

²*School of Mechanical Information, Yiwu Industrial & Commercial College, Yiwu 322000, China*

Correspondence should be addressed to Qi An; anqi@ecust.edu.cn

Received 18 September 2020; Revised 18 October 2020; Accepted 22 October 2020; Published 3 November 2020

Academic Editor: Francesco Riganti-Fulginei

Copyright © 2020 Pengfei Zheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The packing for two-dimensional irregular graphics is one of the NP-complete problems and widely used in industrial applications. In this paper, a descending nesting algorithm for a two-dimensional irregular graph based on geometric feature points is proposed. Before the packing, the parts to be packed are sorted, matched, and spliced, and the matching of the rectangular pieces and the rectangular-like pieces is carried out according to the plate size. On this basis, the geometric feature points of the parts are used to construct the packing baseline, and the packing is accurately carried out according to the principles of the bottom left, the principle of the lowest center of gravity, and combination with virtual moving, rotating collision calculation. The computation of the moving collision distance between the graphics is replaced by the projecting computation of the geometric feature points of the graphic parts, so the computation amount can be reduced. Also, this method is used to test a number of benchmarks examples which are provided by ESICUP (EURO Special Interest Group on Cutting and Packing), which show that the proposed algorithm not only can improve packing but also has better stability and reliability.

1. Introduction

Two-dimensional packing is a classic problem, and the purpose of it is to find a layout with minimal waste on a given shape and size of a plate. The packing technology has been widely used in sheet metal, clothing, shipbuilding, and furniture industries [1]. For the problem of cutting and packing of irregular parts, how to obtain the applicable layout of the geometric conditions is very complicated since all parts must be completely inside the plate and cannot overlap [2]. Two-dimensional irregular packing is a kind of combinatorial optimization problems, which is a NP-complete problem. The calculation complexity is extremely high, and there is no effective mathematical calculation method up to now. Due to its wide application in modern industry, researchers have done a lot of research on this and also proposed many feasible solutions. For example, Baldacci et al. [3] used Lagrangian relaxation method and heuristic algorithm to carry out defect layout and applied it to the leather clothing and furniture industry, but the convergence

of this method is difficult to guarantee. To reduce the complexity of the packing problem, pairwise clustering was introduced in literature [4] to group congruent polygons together in pairs. Pairwise clustering was done automatically to discover matched features among multiple present polygons. In literature [5], a systematic study of the problem and an exact branch-and-bound Algorithm was made by reducing the size of the problem to be solved at each node. Based on the literature [5], Wang et al. [6] utilized a novel branching scheme to deal with the reverse convex quadratic constraints in the quadratic model and incorporated a number of problem-specific algorithmic tweaks. Literature [4–9] all belonged to the NFP- (no-fit polygon-) based packing algorithm. Although some improvements had been made to achieve packing, due to the large amount of calculation of NFP itself, the algorithm efficiency could not be improved well. Based on two known techniques, the vector graphics were transformed into pixel matrix by AutoCAD before packing, and the bottom-left-fill algorithm was used for packing, an irregular stock cutting system (NEST)

developed on AutoCAD software platform, which is presented in [10]. But this method lacks accurate collision detection, and the material utilization rate is not very high. Since the existing packing algorithms do not consider the shape contour features, an irregular packing algorithm was proposed to solve this problem based on a principal component analysis methodology in [11], but the collision detection between two pixels increases the amount of calculation. Jones et al. [12] introduced a fully general, exact algorithm for packing irregular shapes based on a specialized branch-and-bound algorithm together with some heuristics. Camacho et al. [13] proposed an adaptation to the two-dimensional irregular bin packing problem of the Djang and Finch heuristic (DJD) which is originally designed for the one-dimensional bin packing problem. A new mixed-integer model in which binary decision variables were associated with each discrete point of the board (a dot) and with each piece type was proposed in [14]. Based on [14], a mixed-integer-linear programming formulation was proposed to optimize the packing of particular polygonal shapes [15]. These algorithms above used discrete methods to calculate the collision distance, but the discretization method required a large number of data points. Liao et al. [16] adopted a physical method that used the rubber band packing algorithm to simulate a rubber band wrapping those packing irregular shapes. The simulation showed a visual and fast packing process. However, this method had no advantage for mixed packing of multiple irregularities. There are some strip packing algorithms, heuristic algorithms, hybrid algorithms, three-dimensional packing algorithms, and some applications of these algorithms [17–25]. But some are only suitable for rectangular packing, and the calculation is complicated, which will be improved in this paper. In addition, similar heuristic optimization layout ideas have also been applied to more and more fields, which are also worth learning [26–29].

In summary, the two-dimensional irregular graphics packing algorithm can be roughly divided into two categories: one is to simplify the two-dimensional irregular layout problem to a regular piece (rectangular bounding box) layout problem and perform simplified layout and the other one is based on the BL principle, through extrusion, abutment, and pattern rotation to adjust the abutment angle to optimize packing and improve material utilization. However, when you rotate and adjust the docking angle if you choose a preset angle for docking, you will lose the best angle docking. And, in the process of horizontal moving docking and vertical moving docking, it is often to cause the overlapping phenomenon in the previous moving direction after the second movement.

In view of these defects, a two-dimensional irregular descending geometric optimization packing algorithm based on the geometric feature points of graphs is proposed in this paper. This paper has made some contributions in the following aspects:

- (1) Use the graph normalization preprocessing to sort the layout parts in descending order that overcomes the space waste defect caused by random layout.
- (2) Use variable packing baseline to arrange the layout graphs by collision. Overcome the limitation of layered packing strategy and improve the compactness of packing.
- (3) According to the principle of adaptation, the precise touch between the layout graphs is replaced by the discrete feature points of graphs, which greatly reduces the amount of calculation.
- (4) The adaptive virtual collision computing is carried out to reduce the calculation caused by the graphs' movement, which also avoids the phenomenon that the one-way collision calculation easily causes the failure of finding the best packing position.

2. Methods

This paper proposes a two-dimensional irregular packing algorithm. This method builds an irregular packing baseline based on the geometric feature points of the layout piece. In this paper, the layout of the same parts is processed in advance to improve the priority of the same parts packing. Therefore, the regularity of the similar parts packing is effectively improved. The calculation of the distance between horizontal and vertical collisions and the angle of rotation against collisions can greatly enhance the efficiency of packing and the positioning accuracy of a one-time layout. The calculation of the graphic feature points is used instead of the graphic mobile layout, which realizes virtual packing and can save time. Figure 1 shows the main processes of this algorithm.

2.1. Graphics Preprocessing

2.1.1. Graphic Prenormalization. In order to simplify the packing process and reduce the time of repeated searches, before the packing, it is proposed to encode and store graphics with a unified standard to be packing. The normalization rules are to sort the geometric feature points of all the pieces by arranging clockwise and to store them with the BL (bottom left) point as the starting point, to extract the geometric feature points of the packing parts and save them as the points group table named $obj(i)_{plst}$. As shown in Figure 2, the layout part is a polygon formed by eight points connected in sequence. According to the normalization rules, the order of the eight points stored in $obj(i)_{plst}$ is $P_8 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_7$. The purpose of such normalization is to facilitate the subsequent search of the reference points for packing; that is, the first point of each points' group of the part is used as the reference point for the packing movement.

FIGURE 1: Flowchart of the proposed algorithm.

FIGURE 2: Increase graphic feature points. (a) Graphics before midpoint adding process. (b) Graphics after midpoint adding process.

In order to ensure a unique correspondence in the process of matching the initial layout patterns, during the normalization process, the horizontal and vertical lines will automatically add points. As shown in Figure 2(a), where the lines $\overline{P_1P_2}$, $\overline{P_3P_4}$, $\overline{P_5P_6}$, and $\overline{P_7P_8}$ are horizontal lines, and the lines $\overline{P_1P_8}$, $\overline{P_2P_3}$, $\overline{P_4P_5}$, and $\overline{P_6P_7}$ are vertical lines, adding point processing is required for them. The method of increasing the midpoints is adopted. After processing, it is shown in Figure 2(b). Its role is to increase the number of discrete points in order to minimize the loss of information caused by the reduction of the vector graph to the discrete feature points representation and ensure the effectiveness of subsequent graphic matching and collision calculation. In order to reduce the recognition of the graphic features for packing, when the layout pieces contain arc features, the arc discretization will be used to divide the arc into discrete points and connect the discrete points with polylines for approximate replacement. After the packing is finished, the arc features are restored.

2.1.2. Matching Stitching of Graphics. Since the two similar pieces can be reassembled into a more compact new piece, reassembling the similar pieces first and then arranging will greatly save the layout space and improve the utilization of the layout. In the packing process, in order to avoid similar parts as independent pieces to be individually packed, the graphics are matched and stitched during the preprocessing of the graphics. The principle of graphic matching is the preliminary arrangement of packing parts according to area and number of feature points; that is, two layout parts with the same area and the same number of feature points are treated as similar pieces, and the features with the highest degree of coincidence are used as matching objects. As shown in Figure 3(a), the highest coincidence characteristics of the layout pieces P and T are $\overline{P_2P_3}$ and $\overline{T_1T_3}$, respectively. After the matching and stitching process, the layout pieces P and T are merged into a rectangular piece. The prepacking pieces in Figure 3(b) are not the same kind, but they belong to similar kinds. They need to be matched and spliced in the preprocessing stage; otherwise, it will waste the packing space in the subsequent packing. When the moving parts P and T are matched by moving and touching, the feature point P_3 corresponding to the touching points are T_2 and T_3 , and there is a nonsingle correspondence relationship. At this time, such a point is not used as the touching feature point. The same is for points P_2 , P_4 , P_5 , and P_6 . By adding points as described in Section 2.1.1, we can use the newly added midpoints P_{14} , P_{15} , and P_{16} as horizontal touchpoints for matching and stitching. In this way, we can complete the precise splicing of two layout parts and reduce unnecessary waste for the next optimization of packing.

In order to better adapt to the size of the board for packing, during the preprocessing of the graphics, the rectangular parts and rectangular-like parts are

preferentially matched and stitched. As shown in Figure 3(c), the packing piece A is located in the smallest rectangular enclosing frame B . We define the approximation coefficient of a rectangular-like piece as $s_\delta = s_A/s_B$, where s_A is the area of piece A and s_B is the area of piece B . The value of s_δ will directly affect the packing effect. Through a large number of experiments, an appropriate value of s_δ is chosen as follows: if it is satisfied $s_\delta \geq 85\%$, then, A is called a rectangular-like piece. In the preprocessing stage, if a certain piece meets $s_\delta \geq 90\%$, it will be matched and spliced. The maximum length of the splicing is the width of the plate. The purpose is to make the bottom layer “fully packed” as much as possible to reduce the waste of the plate.

2.1.3. Constructing Packing Baseline Based on Geometric Feature Points. In 2D regular parts packing, the strip model is commonly used to achieve layered packing. However, in irregular packing, it is difficult to form a clear strip model and the horizontality of the layered layout cannot be guaranteed, so this strategy is not suitable for the packing of two-dimensional irregular parts. However, the layout idea still has guiding significance. Based on this idea, the strategy of layering layout of the strip model is extended, and the idea of packing baseline based on geometric feature points is proposed. The packing baseline is not a traditional horizontal baseline, but a multipolyline and contour line updated in real time.

Definition 1. As shown in Figure 3(d), we divide the area around a point into four parts: top left, top right, bottom left, and bottom right. We set this point P_i as the eye. If you cannot see that side, then this side is considered closed; if you can see it, this side is considered open, so the top right side of point P_i is close, and other sides are open. If all sides of a point (top left, bottom left, top right, and bottom right) are closed, that is, the point is surrounded by the fixed pieces, these geometric feature points are called closed points, as C_1 and C_2 shown in Figure 4;

Definition 2. If at least one direction around the point (top left, bottom left, top right, and bottom right) is not closed, that is, the point is not completely surrounded by the fixed parts, these geometric feature points are called open points, as O_1 and O_2 shown in Figure 4.

Connecting all the current open points into a polyline in a certain order, that is, the packing baseline based on the geometric feature points, the feature points that make up the baseline are called the open points group and recorded as `base_open_plst`. Similarly, record all the points forming a closed-loop as `close_plst`. As shown in Figure 4, $O_5 \rightarrow C_1 \rightarrow C_2 \rightarrow O_5$ is a closed points group, which is called the packing closed area. The packing closed areas are also the source of waste of layout materials. The smaller the

FIGURE 3: Graphic matching and stitching. (a) Stitching of similar pieces. (b) Stitching of heterogeneous pieces. (c) Rectangular-like piece. (d) Open point and close point.

FIGURE 4: Packing baseline with geometric feature points.

packing closed area, the better the layout effect and the higher the material utilization rate.

As shown in Figure 4, the current packing baseline is \$O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow \dots \rightarrow O_{15}\$, and the currently scheduled piece is A. According to the BL principle, its initial position is the position of the dashed box in Figure 4. Obviously, this position is not the optimal layout, so the position needs to be adjusted by the assistance of a further heuristic collision strategy.

2.2. Packing Rules

2.2.1. Sorting Descending Order. In the graphic preprocessing, all parts have been normalized and matched and stitched, but the matching and stitching process will reorganize some of the layouts (shown in Figure 3), so we need to do a stitching process for the recombinant pieces before packing. The processed parts are used as the initial input graphics. The area of each recombinant part is calculated and packed in descending order according to their area.

According to Pick's theorem, we can transform \$n\$ polygons into \$n-2\$ triangles and use the cross multiplication to calculate the area of each triangle in turn. The sum of the areas of triangles is the area of the polygon. The formula for calculating the area of the prepacking piece (polygon) is

$$S_A = \frac{\sum_{i=1}^n \begin{bmatrix} x_i & y_i \\ x_{(i+1)} & y_{(i+1)} \end{bmatrix}}{2}, \quad (i = 1, \dots, n), \quad (1)$$

where \$x_i\$ and \$y_i\$ are the coordinates of each vertex of part A.

The calculation formula for the centroid of the prepacking piece is

$$x_c = \frac{\sum_{i=1}^n x_i}{n}, \quad y_c = \frac{\sum_{i=1}^n y_i}{n} \quad (i = 1 \dots n), \quad (2)$$

where \$S_A\$ is the area of part A and \$x_c\$ and \$y_c\$ are the centroid coordinates of part A.

The algorithm of the packing process is as follows:

Step 1: calculate the areas of all the prepacking pieces (\$S_A\$), total length (\$L_A\$), and centroid coordinates (\$x_c, y_c\$), extract feature points list (obj(i)_plst), and save them into the data list obj(i)_info_lst(\$S_A, L_A, (x_c, y_c)\$, obj(i)_plst).

Step 2: sort the data list obj(i)_info_lst(\$S_A, L_A, (x_c, y_c)\$, obj(i)_plst) in descending order by \$S_A\$.

Step 3: extract the feature points of the initial layout and record as base_open_plst(\$p_{lt}, p_{lb}, p_{rb}\$). Among them, \$p_{lt}\$ is the upper left corner point of the plate, \$p_{lb}\$ is the lower left corner point of the plate, and \$p_{rb}\$ is the lower

right corner point of the plate. Set the initial closed points list as `close_plst`.

Step 4: take out a piece from the prepacking pieces information list as `obj(i)_info_lst` in order, and record it as the currently active prepacking piece as `active_obj(i)_lst`. Compare the area of the closed area in the current layout with the closed points group. If it can be placed, it will be preferentially located in the closed area. If it cannot be placed, perform virtual mobile packing calculation on the current layout part, virtually move the part to the reference point position of the current packing baseline, and wait for accurate calculation by collision.

Step 5: perform virtual precise collision calculation between the current layout position and the current packing baseline. By repeating the iteration of the calculation, the final layout position is determined, and the coordinates of the final layout position are computed.

Step 6: move the current part from the initial position to the final layout position to complete packing the current part according to the coordinates of the final layout position in the previous step. Calculate the open points group and closed points group of the current layout, add open points group to `base_open_plst`, update, merge, and save the current closed points group into `close_plst`.

Step 7: calculate whether the current part exceeds the plate size; if not, go to Step 4 and continue packing. If it exceeds the plate size, delete the current layout part, go to step 4 to find a suitable prepacking part, until the end of traversing all parts.

2.2.2. Discrete Virtual Packing Calculation. In order to save the time of the moving and contacting test, the geometric feature points of the prepacking piece are extracted as the basis for computing of the virtual packing, and the virtual movement and virtual rotation and contacting test are performed through corresponding calculations. As shown in Figure 5(a), eight discrete feature points of the part are extracted as the feature information for packing. If you need to perform a moving and touching test on a prepacking piece, you only need to calculate the translation matrix of these points to get the new feature points coordinates after the piece moves. Similarly, after calculating the rotation of the centroid of Figure 5(a), the feature points of the virtual rotating arrangement can be obtained as shown in Figure 5(c). In this way, the computation of the contacting test of packing can be reduced, and the operation process of multiple physical translation can be avoided. It only needs to actually move and rotate once after getting the exact position, so this method can achieve precise packing in one step.

The virtual translation calculation formula is

$$\begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (3)$$

where x, y are the original coordinates; x_m, y_m are the coordinates after moving; t_x is the movement in x direction; and t_y is the movement in y direction.

The virtual rotation calculation formula is

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (4)$$

where x, y are the original coordinates; x_r, y_r are the coordinates after the centroid rotation; (t_x, t_y) are centroid coordinates of the prepacking piece; and θ is the rotation angle:

$$\begin{aligned} M &= \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & -\sin \theta & (1 - \cos \theta)t_x + t_y * \sin \theta \\ \sin \theta & \cos \theta & (1 - \cos \theta)t_y - t_x * \sin \theta \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (5)$$

2.2.3. Bottom Left and Center of Gravity Principles and Packing Priorities. In the virtual packing calculation, the BL combined with the lowest center of gravity principle is used as the basis for determining the optimal packing. That is, in the virtual translation packing calculation, try to arrange the layout pieces to the leftmost and bottommost position of the plate, use the packing baseline as the benchmark, and try to squeeze it to the left and bottom. In the calculation of the virtual rotation packing, the rotation angle corresponding to the lowest position of the centroid is used as the angle value required for the optimal layout position.

At the same time, material utilization is also an important indicator for evaluating the effect of packing. According to the principle of optimal packing for maximum plate utilization, the following layout priorities are set.

- (1) The packing process is from left to right and from bottom to top. That is, when the heights of the center of gravity of the two prepacking pieces are equal, the left one takes precedence, and when the distances of the two prepacking pieces from the left boundary are equal, the lower height of the center of gravity takes precedence.
- (2) The ranking of recombinant components priority over the area descending principle.

FIGURE 5: Virtual translation, rotation computation of graphics. (a) Original graphic. (b) Graphic after translation. (c) Graphic after rotation.

- (3) The principle that the highest point is lower after the part is placed will take precedence over the principle of the lowest center of gravity.
- (4) According to the area descending principle, when the current part is placed and exceeds the highest point of the current packing baseline if it meets the plate utilization condition, it is allowed to continue packing once again. If the above situation is not caused by the principle of descending area, there is no such priority.

Plate utilization conditions: $\sum_{i=1}^n s_i \geq s_{\text{over}} = l_{\text{width}} \times h_{\text{over}}$, where $\sum_{i=1}^n s_i$ is the sum of the areas of the remaining pieces to be arranged, s_{over} is the overflow area caused by the current packing, l_{width} is the width of the plate, and h_{over} is the height difference between the highest point after the current packed part and the highest point of the current packing baseline.

- (5) If there are only two pieces to be packed at present, the principle of the lower highest point after the piece is placed is a priority to any other principles.

2.3. Adaptive Precise Collision Computing

2.3.1. Moving and Collision Computing. In order to achieve two-direction precise moving collision packing, a variable step size for horizontal moving layout is used, and then vertical moving and collision calculation is carried out to obtain the lowest collision position. As shown in Figure 6, the leftmost point of the bounding box of piece A is used as the initial position of horizontal movement; then, take accumulative horizontal movement with appropriate steps ($\delta = 0.5 \times \text{box_width}$), take vertical collision calculations at each moving position, and compute the horizontal position corresponding to the minimum value of the observation point P_{bl} of the packing piece (position A as shown in Figure 7). Use this position as the positioning reference for subsequent precise movement. Increase the width of the enclosing frame by 0.5 times to the left and right as the range of accurate horizontal movement. Shorten the step size and recalculate P_{bl} . This method from coarse positioning to precise subdivision collision computing greatly reduces the computation for uniformly discrete and traversal collision. Moreover, in the vertical collision

calculation, the calculation of the distances between the linear discrete points is replaced by the projection calculation of several geometric feature points face to face, which greatly reduces the time of the vertical collision distance calculation. In the calculation of the vertical collision distance, a “face-to-face” collision test is carried out by using a bounding box projection of prepacking piece to intercept a part of the packing baseline to reduce the computation.

There are two types of collision detection calculations: vertical collision detection and horizontal collision detection. The specific detection and calculation processes are as follows: as shown in Figure 7, when part A to be scheduled moves to the location to be scheduled (black line frame in Figure 7), the key points (p_1, p_2, p_3, p_4) of part A are projected onto the current packing baseline ($O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow \dots \rightarrow O_{10}$) for the vertical collision detection; the projection points (t_1, t_2, o_3, t_3) are obtained, respectively; and the projection distances are $\overline{p_1 t_1}, \overline{p_2 t_2}, \overline{p_3 o_3}, \overline{p_4 t_3}$, respectively. Since the “face-to-face” collision detection is mutual, the back projection is performed in the same way, intercept the feature points (o_2, o_3, o_4) located on the packing baseline between the key points of part A, and project these feature points onto part A. The projected points (t_4, t_4, t_5) are obtained, respectively, and the projection distances are $\overline{o_2 t_4}, \overline{o_3 t_4}, \overline{o_4 t_5}$, respectively. Take the shortest distance of the vertical collision detection as the vertical movement amount, and record as L_v .

In Figure 7, $L_v = \min\{\overline{p_1 t_1}, \overline{p_2 t_2}, \overline{p_3 o_3}, \overline{p_4 t_3}, \overline{o_2 t_4}, \overline{o_3 t_4}, \overline{o_4 t_5}\} = \overline{o_2 t_4}$, there is only one tough point for part A to be arranged, so the current layout position is not completely touched; it needs to cooperate with the horizontal collision detection for further collision computing.

In the calculation of the horizontal collision detection, the process is similar to the vertical collision detection, and it also uses two-way mutual projection to compute the shortest projection distance.

As shown in Figure 8, the movement amount of horizontal collision is $L_h = \overline{p_4 t_6}$. Because the horizontal collision movement will cause the original vertical tough relationship to be invalidated, in order to automatically and repeatedly detect the vertical and horizontal collision and finally reach the best tough position, a suitable margin is set for the movement

FIGURE 6: Graphic moving collision computation.

FIGURE 7: Graphical computing of vertical collision detection.

amount of the horizontal collision. After tests, when the actual horizontal movement amount is $L'_h = 0.5L_h$, it is beneficial to increase the speed of collision detection. After repeating vertical and horizontal collision detection calculations, the final packing position of part A is A3, as shown in Figure 9.

2.3.2. Rotating and Collision Computing. When computing the lowest layout position, the 90° rotation angle is used for the general rectangular and rectangular-like parts ($s_\delta \geq 85\%$) to compute the collision position, and for nonrectangular parts, use any angles to rotate and contact. As shown in Figure 9, A3 is the best position for the two-direction movement after collision, but it is not the lowest position. It can rotate piece A around its centroid, then use vertical and horizontal collision detection, and find the lower position A4 as the final layout position. In order to save the time of packing, the above movements and rotations are performed in the form of virtual movement and rotation computing.

3. Results

In order to further verify the effectiveness of this proposed algorithm, the algorithm is tested on a general auxiliary

design platform AutoCAD. The benchmark problems provided by the ESICUP (EUROSICUP) are used as test cases. Some commonly used benchmark examples, blaz, marques, shirts, trousers, and so forth, are chosen for the packing test. The packing results are shown in Figures 10–13.

4. Discussion

In order to facilitate comparison, the analysis results of this method, the literature [30, 31], and commercial packing software of SigmaNest and NestLib are shown in Table 1 (packing density is the ratio of the sum of the area of the parts to the area of the plate).

It can be seen from Table 1 that the descending order packing algorithm is reliable. After comparison with SigmaNest, NestLib, GEFHNA, HAPG_HRG, and other methods, the conclusion is drawn: irregular graphics descending packing algorithm based on geometric feature points is suitable for the problem of two-dimensional irregular graphics packing. The packing density is high, and it has obvious advantages in dealing with the problem of mixed layout of regular and irregular pieces. The algorithm is

FIGURE 8: Graphical computing of horizontal collision detection.

FIGURE 9: Graphic rotating collision computation.

FIGURE 10: The packing result of our algorithm (blaz).

FIGURE 11: The packing result of our algorithm (marques).

FIGURE 12: The packing result of our algorithm (shirts).

FIGURE 13: The packing result of our algorithm (trousers).

TABLE 1: Computing results of examples (packing density).

Case	SigmaNest (%)	NestLib (%)	Reference [30] (%)	Reference [31] (%)	Ours (%)
Blaz	67.84	69.68	74.06	—	75.79
Marques	75.03	83.15	77.72	80.43	85.85
Dagli	77.97	76.84	79.41	—	79.41
Fu	77.85	73.08	78.71	—	79.17
Jakobs1	67.96	75.38	75.38	—	75.38
Jakobs2	59.92	64.33	68.93	—	68.93
Shapes0	54.25	57.83	57.83	—	57.83
Shapes1	54.17	61.38	59.11	—	59.18
Shirts	77.37	77.14	79.13	68.92	80.10
Trousers	82.73	82.50	80.78	84.85	83.53

general, simple to operate, and has good application economics.

5. Conclusions

A general geometric packing algorithm suitable for two-dimensional irregular graphics is proposed in this paper. The packing order is guided by the descending order of the area of parts. Through the graphic preprocessing, rectangular pieces and rectangular-like pieces are automatically matched and spliced according to their sizes, which eliminate the waste of space caused by the single packing of pieces and improves the sheet utilization. The packing baseline is constructed by the geometric feature points of the layout parts, prepacking parts are moved according to the BL principle, and the geometric feature points of the part and the packing baseline are used as the key point for bidirectional collision computing. The portion of feature points of the packing baseline are intercepted for face-to-face calculation by way of the bounding box of the layout piece, which greatly reduce the traversal computation cost of the method of converting vector graphics to discrete points. The computing of the lowest packing position of parts is realized by using large-step positioning and small-step accurate collision computation, which avoids the traversal computing of NFP method.

This paper has made some contributions in the following aspects:

- (1) This algorithm classifies graphics according to their geometric characteristics and performs graphic classification and recombination preprocessing before packing, which solves the problems caused by single packing of previous algorithms.
- (2) The adaptive vertical collision detection and horizontal collision detection methods can find more accurate and compact layout positions, which is an innovation of this paper.
- (3) The strip layout algorithm is extended, the layered packing strategy is extended, and the strategy of variable packing baseline is used to improve the compactness of the layout.

Future research will be carried out in the following aspects: optimize the algorithm for computing the packing baseline and rotation collision to further improve the

screening efficiency of the initial packing of the layout pieces; design the adaptive adjustment method of the rotation angle of the graphics, and use the closed free spaces for packing. A better mechanism for rational selection of the layout order based on the principle of highest plate utilization will be studied. The idea of using low-dimensional computing of discrete points of geometrical graphic items to solve complex problems of measurement and positioning between geometric graphics will be applied to more engineering fields, and the approaching of geometrical computing better for solving geometry problems will continue to be promoted.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- [1] W.-C. Lee, H. Ma, and B.-W. Cheng, "A heuristic for nesting problems of irregular shapes," *Computer-Aided Design*, vol. 40, no. 5, pp. 625–633, 2008.
- [2] A. K. Sato, T. C. Martins, and M. S. G. Tsuzuki, "An algorithm for the strip packing problem using collision free region and exact fitting placement," *Computer-Aided Design*, vol. 44, no. 8, pp. 766–777, 2012.
- [3] R. Baldacci, M. A. Boschetti, M. Ganovelli, and V. Maniezzo, "Algorithms for nesting with defects," *Discrete Applied Mathematics*, vol. 163, pp. 17–33, 2014.
- [4] A. Elkeran, "A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering," *European Journal of Operational Research*, vol. 231, no. 3, pp. 757–769, 2013.
- [5] R. Alvarez-Valdes, A. Martinez, and J. M. Tamarit, "A branch & bound algorithm for cutting and packing irregularly shaped pieces," *International Journal of Production Economics*, vol. 145, no. 2, pp. 463–477, 2013.
- [6] A. Wang, C. L. Hanselman, and C. E. Gounaris, "A customized branch-and-bound approach for irregular shape nesting," *Journal of Global Optimization*, vol. 71, no. 4, pp. 935–955, 2018.
- [7] R. P. Abeysooriya, J. A. Bennell, and A. Martinez-Sykora, "Jostle heuristics for the 2D-irregular shapes bin packing

- problems with free rotation," *International Journal of Production Economics*, vol. 195, pp. 12–26, 2018.
- [8] A. K. Sato, T. d. C. Martins, and M. d. S. G. Tsuzuki, "A pairwise exact placement algorithm for the irregular nesting problem," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 11, pp. 1177–1189, 2016.
 - [9] P. R. Pinheiro, B. Saraiva, and R. D. Saraivab, "A random-key genetic algorithm for solving the nesting problem," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 11, pp. 1159–1165, 2016.
 - [10] W.-C. Weng and H.-C. Kuo, "Irregular stock cutting system based on AutoCAD," *Advances in Engineering Software*, vol. 42, no. 9, pp. 634–643, 2011.
 - [11] B. Guo, Z. Liang, Q. Peng, Y. Li, and F. Wu, "Irregular packing based on principal component analysis methodology," *IEEE Access*, vol. 6, pp. 62675–62686, 2018.
 - [12] D. R. Jones, "A fully general, exact algorithm for nesting irregular shapes," *Journal of Global Optimization*, vol. 59, no. 2-3, pp. 367–404, 2014.
 - [13] E. L. Camacho, G. Ochoa, H. T. Marín, and E. K. Burke, "An effective heuristic for the two-dimensional irregular bin packing problem," *Annals of Operations Research*, vol. 206, pp. 241–264, 2013.
 - [14] F. M. B. Toledo, M. A. Carravilla, C. Ribeiro, J. F. Oliveira, and A. M. Gomes, "The Dotted-Board Model: a new MIP model for nesting irregular shapes," *International Journal of Production Economics*, vol. 145, no. 2, pp. 478–487, 2013.
 - [15] M. C. Santoro and F. K. Lemos, "Irregular packing: MILP model based on a polygonal enclosure," *Annals of Operations Research*, vol. 235, no. 1, pp. 693–707, 2015.
 - [16] X. P. Liao, J. Y. Ma, C. Y. Ou, F. Y. Long, and X. S. Liu, "Visual nesting system for irregular cutting-stock problem based on rubber band nesting algorithm," *Advances in Mechanical Engineering*, vol. 8, no. 6, pp. 1–15, 2016.
 - [17] M. Chen, K. Li, D. Zhang, L. Zheng, and X. Fu, "Hierarchical search-embedded hybrid heuristic algorithm for two-dimensional strip packing problem," *IEEE Access*, vol. 7, pp. 179086–179103, 2019.
 - [18] S. Martello and M. Monaci, "Models and algorithms for packing rectangles into the smallest square," *Computers & Operations Research*, vol. 63, pp. 161–171, 2015.
 - [19] H. Chen, F. R. Ye, and Y. W. Si, "A hybrid algorithm for strip packing problem with rotation constraint," in *Proceedings of the MATEC Web of Conferences*, pp. 1–5, Shanghai, China, October 2016.
 - [20] Z. Chen and J. Chen, "An effective corner increment-based algorithm for the two-dimensional strip packing problem," *IEEE Access*, vol. 6, pp. 72906–72924, 2018.
 - [21] Y. J. Li and S. Y. Ji, "A geometric algorithm based on the advancing front approach for sequential sphere packing," *Granular Matter*, vol. 20, pp. 20–59, 2018.
 - [22] S. D. Filonov, K. A. Aksyonov, and V. N. Kruglov, "Method for loading cargo trucks using two-dimensional packing algorithms," *Journal of Physics: Conference Series*, vol. 1210, 2019.
 - [23] M. Chen, C. Wu, X. Tang, X. Peng, Z. Zeng, and S. Liu, "An efficient deterministic heuristic algorithm for the rectangular packing problem," *Computers & Industrial Engineering*, vol. 137, p. 106097, 2019.
 - [24] A. Grange, I. Kacem, and S. Martin, "Algorithms for the bin packing problem with overlapping items," *Computers & Industrial Engineering*, vol. 115, pp. 331–341, 2018.
 - [25] S. P. Fekete, S. Morr, and C. Scheffer, "Split packing: algorithms for packing circles with optimal worst-case density," *Discrete & Computational Geometry*, vol. 61, no. 3, pp. 562–594, 2019.
 - [26] W. Deng, J. J. Xu, Y. J. Song, and H. M. Zhao, "An effective improved co-evolution ant colony optimization algorithm with multi-strategies and its application," *International Journal of Bio-Inspired Computation*, vol. 5, pp. 1–10, 2020.
 - [27] W. Deng, J. J. Xu, H. M. Zhao, and Y. J. Song, "A novel gate resource allocation method using improved PSO-based QEA," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
 - [28] W. Deng, H. Liu, J. Xu, H. Zhao, and Y. Song, "An improved quantum-inspired differential evolution algorithm for deep belief network," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7319–7327, 2020.
 - [29] W. Deng, J. J. Xu, Y. J. Song, and H. M. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Applied Soft Computing*, pp. 1–16, 2020.
 - [30] D. Y. Tang and Z. L. Zhou, "No-fit-polygon-based heuristic nesting algorithm for irregular shapes," *Journal of Computational and Applied Mathematics*, vol. 36, pp. 2540–2544, 2016.
 - [31] X. Liu, J. W. Ye, and J. M. Liu, "Two-dimensional constructive packing algorithm based on hybrid representation graphics," *Journal of Shanghai Jiaotong University*, vol. 52, pp. 825–830, 2018.