Sequence Transfer-Based Particle Swarm Optimization Algorithm for Irregular Packing Problems

JIE FANG, YUNQING RAO, PAN LIU, AND XUSHENG ZHAO
State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
Corresponding author: Yunqing Rao (ryq@hust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51975231, and in part by the Foundational Research Funds for the Central Universities under Grant 2019kfyXJJC043.

ABSTRACT The two-dimensional (2D) irregular packing problem is a classical optimization problem with NP-hard characteristics and has high computational complexity. To date, packing problems have generally been solved by artificial experience and heuristic algorithms. However, these algorithms are not highly efficient and the excellent cases cannot be preserved, which both time and economic costs are high. Inspire by transfer learning and considering the characteristics of 2D irregular packing problems, we propose a sequence transfer-based particle swarm optimization algorithm (ST-PSO) to solve the multi-constraint packing problem. A piece-matching strategy based on an improved shape context algorithm, and a piece-sequencing generation strategy for transferring the packing sequence are developed for particle swarm optimization (PSO) initialization. In the process of PSO, an adaptive adjustment strategy is used with an improved positioning strategy to adjust the packing position of the pieces. The results indicate that this method can robustly, quickly, and efficiently achieve the packing of 2D irregular pieces. Compared with the data prior to transfer, the ST-PSO can inherit and transfer the historical packing sequence in less time and retain or exceed the actual packing data onto the samples. This algorithm could be applied to industrial applications to reduce waste, packing time, and production costs.

INDEX TERMS Irregular pieces, multiple constraints, particle swarm optimization algorithm, sequence transfer, two-dimensional packing optimization.

I. INTRODUCTION
Packing optimization problems generally exist in the production process of plate metals, steel structures, ships, clothing, leather and paper products, glass, and other industries. Such problems exist in various key steps, from design to plate cutting, in the manufacturing automation processes [1]. In practical applications, such as heavy industrial enterprises, an effective packing algorithm reduces calculation time, increases resource utilization, and thereby reduces waste (e.g., raw materials) and production costs. The packing problems can be categorized into sub-types such as 1D, 2D and 3D packing. The most classic problem is the optimization of 2D plate packing [2]. The problem of 2D plate packing is to place the pieces, which are to be laid out in the packing space (in a certain manner within the given 2D packing space to satisfy certain criteria) to maximize the utilization rate of the pieces or minimize their total height. The 2D packing problem is an interesting combinatorial optimization problem, as well as an NP-hard problem; hence, it is a very popular research focus in the optimization field [3]. Compared with packing regular 2D pieces, 2D irregular piece packing problems have a larger solution space and more complex packing operations, and the computational complexity increases exponentially with iteration number; thus, an optimal solution of the problem is difficult to obtain. Therefore, many researchers have proposed various approximate or heuristic algorithms from the perspective of sequencing and positioning [4]. Sequence optimization is mainly used to optimize the order in which pieces are placed in the front and back of the piece from a global perspective, to optimize the use of the main plane space occupied by the packing piece, and to increase the
II. PROBLEM STATEMENT

In a typical 2D packing problem, the plate and pieces can have one or more contours, which can be irregular or regular polygons. The packing process aims to minimize the used area or the number of the plates by arranging a group of pieces on the plate, that is, the utilization rate of the plate is maximized, wherein the optimization goals and constraints of the 2D packing will vary depending on the specific criteria of the application. According to the characteristics of a variable batch production mode in typical heavy industries, the width of the plate is fixed, and the plate height is not limited. As the cost of a plate is directly proportional to area occupied by the pieces, there are large economic benefits in minimizing the packing height or maximizing the utilization rate of the plates.

A general description of the packing model is given below. Given a rectangular plate, the width of the plate is $W$, the number of packing pieces is $n$, the area of the packing
pieces is $S_i$, and the maximum height of the pieces on the plate after packing is $H$. Here, $U$ is the utilization rate, $z_i$ indicates the packing state of the piece on the plate: if packing can be performed, $z_i = 1$, but if the packing cannot be performed, $z_i = 0$. The position of the irregular 2D piece on the rectangular plate is determined by the coordinate $(x, y)$ of the piece and the discharge angle $\alpha$. The piece can be positioned by applying translations and rotations to the piece. Therefore, the orientation of the piece on the plate can be expressed as $G_i(x'_i, y'_i, \alpha_i)$. Finally, the objective function of optimal packing is expressed as follows:

$$\max U = \max \left\{ \frac{\sum_{i=1}^{n} z_i S_i}{W H} \right\}$$

(1)

The constraints of packing are

$$G_i(x'_i, y'_i, \alpha_i) \cap G_k(x'_k, y'_k, \alpha_k) = \emptyset,$$  

(2)

$$0 \leq x(x'_i, y'_i, \alpha_i) \leq H,$$  

(3)

$$0 \leq y(x'_i, y'_i, \alpha_i) \leq W.$$  

(4)

Formula (2) indicates that no two pieces can overlap, and formulas (3) and (4) indicate that the coordinates of any point in the piece must be within the range of the packing plate, and there must be no piece beyond the plate.

### III. NOVEL POSITIONING STRATEGY BASED ON NFP

#### A. NO-FIT POLYGON

The background concept of the NO-fit polygon method is first proposed by Art [26], this is a process used to avoid overlapping when pieces are placed on a plate. Ten years later, Adamowicz [27] named this method “NO-fit polygon” (NFP). Since then, many researchers have continued to develop and improve NFP. Although different methods of solving the NFP problems have been proposed, they can be summarized into three general methods: (a) Minkowski sums [28]–[31], (b) the decomposition algorithm [32]–[34], and (c) the orbiting algorithm [35], [36].

Given two polygons, when polygon $P_j$ slides around the outer boundary of polygon $P_i$ (fixed), the trajectory of the reference point on polygon $P_j$ forms the NFP $(P_i, P_j)$. During the sliding process, the two polygons do not overlap or rotate, as shown in Figure 1. The three cases regarding the position of the reference point are as follows:

(a) If the reference point is located in NFP $(P_i, P_j)$, then polygons $P_i$ and $P_j$ overlap.

(b) If the reference point is located on the boundary of the NFP $(P_i, P_j)$, then polygons $P_i$ and $P_j$ contact each other.

(c) If the reference point is outside the NFP $(P_i, P_j)$, then polygons $P_i$ and $P_j$ are not in contact.

In general implementation, all possible NFPs are computed in a preprocessing level applying the (a) Minkowski sum method and (b) polygon decomposition. Given two arbitrary point sets, $P$ and $Q$, the Minkowski sum of $P$ and $Q$ is represented as $P \oplus Q = \{p + q; p \in P, q \in Q\}$, which assumes that the vectors of the two polygons are in the same direction.

Suppose $P$ is counter-clockwise and $Q$ is clockwise, then, $P \oplus (-Q)$, which is the Minkowski difference of $P$ and $Q$, is equivalent to NFP $(P, Q)$ since a stipulation that polygons stay in counter-clockwise orientation, $-Q$, is simply transposed to symmetric vector of the $Q$. The concept of NFP applicable only for convex polygons. However, the application of non-convex polygons has a higher complexity, and the main solutions for it are Minkowski sums, decomposition in several sub-polygons and orbit sliding algorithms. The orbit sliding algorithm involves the use of standard trigonometry to physically slide a polygon, and the NFP is defined by tracking the movement of the trajectory point of the sliding polygon on the orbit. The key elements of Mahadevan’s [36] approach to orbit are the calculation of contact vertices and edges, the determination of translation vectors and the calculation of translation length. The disadvantage of the method is that it cannot generate NFP for shapes containing holes or concave surfaces.

In this study, the calculation of the NFP is processed using the track-sliding algorithm, which is similar to the algorithm of Rao et al. [19] in terms of algorithm logic. In the application process, the principle of NFP generated by the sliding algorithm is shown in Figure 1. And the Inner-Fit Polygon (IFP) is similar to the NFP except that the orbiting polygon, in this case, slides around inside the stationary polygon. Given a plate $P$ and a polygon $Q$ with a certain angle of entry, the plate $P$ is defined as a fixed polygon, and the piece $Q$ is defined as a sliding polygon. By sliding $Q$ tangentially along the inner contour of the plate $P$, and the closed polygon formed by the reference point on $Q$ is IFP $PQ$.

#### B. POSITIONING STRATEGY OF HYBRID BL AND LOWEST CENTER OF GRAVITY

The solution space of the irregular pieces packing problem is infinite without limiting the placement position and rotation angle. To reduce the computational complexity and obtain a good packing efficiency, some scholars have proposed placement strategies such as bottom-left (BL), bottom-left fill (BLF), best bin first (BBF), and TOPOS. Among them, the BL algorithm can record all of the current horizontal lines and arrange the pieces in the left and bottom available horizontal lines. In contrast, the TOPOS algorithm uses different evaluation criteria to evaluate the placement of pieces;
however, it is generally suitable for orthogonal packing problems, but not for the packing of irregular pieces.

Here, to determine the placement position and rotation angle of each piece, a strategy based on the BL algorithm and the principle of the lowest center of gravity is proposed. We select two attributes, namely the length $L$ and area $A$ of the rectangle to create criterion $Z$, as shown in the following formulas (5), (6), and (7). Figure 2 presents the definition of the formula.

$$\text{Length : } L = \frac{L_{\text{new}}}{x_{\text{new}}} \tag{5}$$

$$\text{Area : } A = \frac{L_{\text{new}} \times W_{\text{new}}}{x_{\text{new}} \times y_{\text{new}}} \tag{6}$$

$$\text{CostFunction} (\text{criterion}) = \min (L + A) \tag{7}$$

**FIGURE 2.** Expression of the criteria for the new piece.

After $L$, $A$, and $Z$ have been chosen, a piece is selected, and the center of gravity of the piece is first calculated, from which one can generate two NFPs. These are the gravity-center NFP and the bottom-left NFP, generated by viewing the center of gravity or the bottom-left vertex as the reference point, respectively [19]. It is worth noting that the two NFPs will be different when the parts are rotated at different angles. The lowest center of gravity and bottommost-leftmost vertex are chosen to be evaluated by the criteria presented earlier. The position with the best evaluation is determined as the placement location. The placement principle is presented in Figure 3. The two solid polygons, denoted by 1 and 2, are the center of gravity NFPs with two different rotations. The dotted polygons, denoted by 1 and 2, are the bottom-left NFPs.

**FIGURE 3.** Finding the bottommost-leftmost vertex and lowest center of gravity.

IV. METHOD OF SEQUENCE TRANSFER FOR 2D IRREGULAR PIECE PACKING PROBLEMS

A. SEQUENCE TRANSFER

In this section, the method of sequence transfer-based particle swarm optimization is described, combined with the positioning strategy based on the true shape of the piece to solve the 2D irregular piece packing problem. The knowledge-transfer method incorporates knowledge of historical optimal packing sequences to help optimize the placement and angular adjustments of the pieces to make the best use of the free area generated during the process of piece placement.

1) PIECES MATCHING STRATEGY

The two most critical steps in the piece matching are piece feature extraction and piece recognition matching. The shape of the piece graphic can be defined as the contour, where the contour information on an image of the piece is extracted using an edge extraction algorithm. Common feature extraction methods include the scale-invariant feature transfer operator (SIFT) [37], shape context (SC) [38], and interior distance context (IDSC) methods [38]. The process of the shape matching is achieved by reflecting the similarity between the shapes by the similarity between the shapes. The cost matrix is used to retain the information between the shapes, and the similarity is usually measured by the distance between various points in the image. Typical matching methods include IDSC [39], spectral matching [40], multiscale matching [41], linear programming (LP) [42], and dynamic programming (DP) [38]. The shape context method used in this study is first proposed by Mori et al. [43]. This shape-matching algorithm regards the contour of the image as a collection of points, where point-set matching is the key technology. Compared to other algorithms, the shape context method can match the corresponding shape points without special key points, and it is suitable for all ordinary deformations, which can reduce some of the typical errors caused by image preprocessing. The disadvantage is that it is sensitive to noise and requires the boundary points to be consistent with the centroid of the piece, calculating the centroid is time-consuming, and the selected contour points may miss some key points containing image information. Moreover, there is no rotation invariance of the piece graphs. In this section, we describe how these shortcomings are overcome by our method. The piece set matching procedure of source task and target task is given as Algorithm 1. In Algorithm 1, the matching results of pieces in the target task and pieces in the source task (historical task) are stored in the set $MC$, which is an intermediate storage medium, and the corresponding distance values of pieces are stored in the set $DL$. The set $MC$ is emptied since the match starts. The distance values between
the pieces in the target task (new task) set and the pieces in each source task, respectively, are calculated, and the pieces with the smallest distance value from the piece in the corresponding target task are taken as the current matching pieces, which are saved to the corresponding set DI. After the pieces in the target task are matched with the pieces in each source task, the average distance value of each set DI is calculated, and the source task corresponding to the set DI of the current minimum average distance value is taken as the set that matches the target task, and the current set MC and the corresponding piece distance value set DI are saved. It is worth noting that, the optimal packing sequence will be stored in the information storage module (ISM) after the final packing task is completed, in other words, the historically optimal packing sequence is stored in the ISM, which can be used as a source task data set.

**a: IMAGE PREPROCESSING**
In this method, the image is de-noised, and the median filter that performs best in the traditional test is improved. The steps are as follows. First, an initial filter window is set, and then the gray-scale order of each pixel in the window image is traversed and the maximum gray value, minimum gray value, and median value are calculated. Second, if the median value is between the maximum and minimum gray values, and the current pixel gray value is between the maximum gray value and the minimum gray value, then the current gray value is the output; otherwise, the output is in the median value. Third, if the median value is not between the maximum and the minimum gray values, the initial filter window is increased by 2, and the process is repeated from the second step.

**b: EDGE DETECTION**
Here, the Canny operator is used to detect the edge of the image. After the image is binarized, it is first smoothed by Gaussian filtering, and then the gradient amplitude and direction are calculated. The image is processed with non-maximum suppression to enhance the boundary. The gradient direction of the current pixel is approximated in four directions and compared with the size of three pixels in the diagonal direction on the 3 × 3 field. Finally, the edge is detected and connected, and the edge is tracked by chain code.

**c: FEATURE POINT EXTRACTION**
It is not necessary to obtain the centroid coordinates of the boundary points according to the traditional centroid-calculation formula. The distance between the centroids of the original target and the boundary points is measured, and the number of boundary points is determined. According to the number of boundary points from the historical shape context, the range can be estimated as between 50 and 150. However, according to the complexity of the actual pieces and the number of corner points, the fixed value can be set to 100, which saves time by avoiding the repeated comparison and calculation of the number of boundary points. Then, the image can be represented by $$M = (m_1, m_2, m_3, \ldots, m_n)$$, wherein any point $$m_i$$ of the selected piece can form a group of vectors with the remaining $$n - 1$$ points, except this point [44].

As shown in Figure 4, the composed $$n - 1$$ vectors contain a wealth of information, including the relative positions of the remaining points and the reference point, and the remaining $$n - 1$$ vectors in the set can completely represent the entire shape.

**d: LOGARITHMIC POLAR COORDINATES OF SHAPE CONTEXT**
Vector groups constitute the relative positions of point sets, which can be discretized, and the position and shape of the vectors can be described by using log-polar coordinates in the discrete space, which refers to the coordinate system. Generally, Cartesian coordinates $$(x, y)$$ can be used to record the position of each pixel in the image, and polar coordinates $$(r, \theta)$$ can also be used to describe the pixel position. The points in the Cartesian coordinate system are mapped to the polar coordinate system $$(r, \theta)$$ through the transformation of formula (8) and (9). Dividing the logarithmic-polar-coordinate system into 12 equal parts and then dividing each part into five further parts with $$\log_2^5$$, the contour points around it are mapped to each area, and the number of contour points that fall within each area is then counted. Focusing on the characteristic that the traditional shape context has no rotational invariance of the piece graphics, this study proposes a method to correct the angle of a point when calculating the shape context of the point. It takes the direction of the line between the point and the centroid as the polar axis, so that matching the piece graphics is not affected by the piece angle.

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (8)$$
$$\theta = \arctan\left(\frac{y - y_0}{x - x_0}\right) \quad (9)$$

**e: BIPARTITE GRAPH MATCHING**
Bipartite graph matching is also called the square distribution problem, which obeys $$\chi^2$$ distribution, because the matching is a one-to-one correspondence, and the shape context is originally represented by a histogram. For any two points $$p_i$$ and $$q_i$$, located in the two pieces, respectively, it is necessary to calculate the similarity between the two points, that is, the matching cost, to evaluate whether the two points match. The matching cost is defined by the expression

![FIGURE 4. Vector image corresponding to the letter A and point set.](image-url)
\[ C_{ij} = C(p_i, q_j), \] which is defined by formula (10).

\[ C_{ij} = C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \left( \frac{h_i(k) - h_j(k)}{h_i(k) + h_j(k)} \right)^2 \]  

(10)

Here, \( h_i(k) \) and \( h_j(k) \) are the normalized \( k \)th level histograms of points \( p_i \) and \( q_j \), respectively [45]. The smaller the value of formula (10) is, the more closely \( p_i \) and \( q_j \) are connected, and the better the similarity will be. For two different piece shapes, the contour point of one shape can be used as a reference, and the matching cost of the contour point on the other shape can be calculated circulatively to complete the matching. The purpose of piece matching is to find a replacement matching to find the matching point of the node once; and (ii) if the node has been matched, the only way to reach this node through the augmented path. There are two issues of note for the Hungarian algorithm: (i) each node can only be used as the starting point of the augmented path once; and (ii) if the node has been matched, the only way to reach this node through the augmented path is to reach the matching point of the node first.

\[ H(\pi) = \sum_i C(p_i, q_{\pi(i)}) \]  

(11)

**FIGURE 5.** Schematic diagram of a bipartite graph.

\( f: \text{TIN PLATE SPLINE MODEL TRANSFORMATION} \)

The thin plate spline (TPS) model is often used to describe thin metal sheets, where the interpolation method looks for a smooth surface with the smallest curvature through all control points, as defined by an energy function. The TPS model can determine the difference between two pieces by the deformation between them. Almost all deformation can be simulated by TPS. To match the point sets of two pieces, it is necessary to define the function of bending energy to be minimized. To achieve this, it is necessary to select the appropriate \( T \) from a series of transformations. The standard method is the use of a radial transformation, \( T(x) = A(x) + o \), where \( A \) is the matrix and \( o \) is the offset.

\( g: \text{SHAPE DISTANCE AND SIMILARITY CALCULATION} \)

Herein, the distance with shape context and the weighting with bending energy are used to calculate the shape distance to evaluate the similarity between piece images. The sum of the cost of shape context matching is calculated symmetrically between all the best matching points of \( P \) and \( Q \), as the shape context distance. The formula is shown in (12). After point matching, the apparent distance of the figure contour can be obtained, as shown by formula (13), which is defined as the sum of squared brightness difference in Gaussian windows around corresponding image points. After the thin spline transformation \( T \) distorts the image to the best match, we sum over squared difference in windows around corresponding points, scoring the weighted gray-level similarity. The pseudocodes of the algorithm are shown in Algorithm 2, where \( p_1 \) and \( p_2 \) are the piece images in the source task and target task, respectively, while \( M_s \) and \( M_t \) are the point set on the piece images of the source task and the target task, respectively. Further, \( C_A \) is the local appearance cost matrix (formula (14)), and \( C \) is the combined matching cost matrix (formula (15)). The bending energy distance \( D_{be} \) can be obtained by the iteration of TPS, which corresponds to the amount of transformation necessary to align the piece shapes. In the case of TPS, the bending energy is a natural measure [46]. Then, the similarity of the two piece images is \( D = 1.6D_{sc} + DI_{sc} + 0.3D_{be} \).

\[ DI_{sc}(P, Q) = \frac{1}{n} \sum_{p \in P} \arg \min_{q \in Q} C(p, T(q)) \]

(12)

\[ D_{sc}(P, Q) = \frac{1}{n} \sum_{i=1}^{n} \sum_{\Delta \in Z^2} G(\Delta) \left| I_p(p_i + \Delta) - I_q(T(q_{\pi(i)}) + \Delta) \right|^2 \]

(13)

\[ C_A = \frac{1}{2} \left( \cos(\theta_1) - \cos(\theta_2) \right)^2 \]

(14)

\[ C = (1 - \beta)C_{ij} + \beta C_A \]

(15)

Here, \( T() \) represents the shape transformation of the estimated TPS. \( I_p \) and \( I_q \) are the gray-level images corresponding to \( P \) and \( Q \), respectively. \( \Delta \) denotes some differential vector

**Algorithm 1 Piece Set Matching of Source Task and Target Task**

1. Initialize the set \( MC \) for pieces matching
2. while traversing the ISM parts of each piece do
3. for each piece do
4. Calculate the distance of pieces between new tasks and historical tasks
5. Save the set \( DI \) of minimum distance and the index of corresponding historical piece
6. if the index of ISM = 1
7. Update the \( d_i \) and the index of the piece in \( MC \)
8. else
9. Assign \( d_i \) and index of the piece in \( MC \) to \(-1\)
10. return \( MC \) and the set \( DI \)
Algorithm 2: Shape Context Distance

1. Import binary image \( p_1 \), use Canny operator to extract image contour
2. Sampling \( n \) points uniformly on the contour, get point set \( M_i = \{m_{i1}, m_{i2}, \ldots, m_{in}\} \)
3. for \( i = 1 \) to \( n \) do
   1. Establish a polar-logarithmic coordinate system centered on \( m_{i} \)
   2. Adjust the direction of the polar axis to point to the centroid of \( p_1 \)
   3. Compute a coarse histogram \( h_i \) of the remaining \( n-1 \) points
4. Import binary image \( p_2 \), repeat 1 to 3, get shape context histogram set \( M_0 = \{m_{01}, m_{02}, \ldots, m_{0n}\} \) of \( p_2 \)
5. Compute shape context cost matrix \( C_{ij} \)
6. For \( p_1 \) and \( p_2 \), calculate the tangential angle of the gray gradient for each sampling point, get angle set \( \theta_1 \) and \( \theta_2 \)
7. Compute local appearance cost matrix \( C_A \)
8. Get combined matching cost matrix \( C \)
9. Use Hungarian method to get correspondences with cost matrix \( C \), output minimum matching cost as the shape distance between \( p_1 \) and \( p_2 \)

offset and \( G \) is a windowing function, and is typically chosen to be a Gaussian, thus putting emphasis to pixels nearby. \( \beta \) is a constant with a value of 0.1.

In Algorithm 2, a method for calculating the degree of difference between the shapes of 2D pieces is described. \( p_1 \) and \( p_2 \) are two binary images of the pieces to be compared respectively, where \( p_2 \) the reference image. Before the shape distance of the two pieces is calculated, it is necessary to sample the points uniformly on the contour points, and a more abstract dot matrix is used to represent the entire piece shape. A total of \( n \) points is uniformly sampled on the contours of \( p_1 \) and \( p_2 \), and corresponding lattice sets \( M_{s1} \) and \( M_{s2} \) are obtained, respectively. Each point in the lattice forms a context relationship with the other \( n-1 \) points. Therefore, a certain block can be divided around each point according to distance and angle, and then the other \( n-1 \) points around the point will be distributed in these blocks, so that the number of points in each block around this point can be counted to form a statistical histogram, which can be called shape context histogram. The shape context histogram can be obtained in the same way for each point in the lattice. If two shapes are similar, the shape context histograms of the points at the same position are similar. By comparing the shape context histograms of each pair of sampling points in the lattice of the two contours with the chi-square statistics method, the distance matrix between all points in set \( M_{s1} \) and set \( M_{s2} \) can be obtained [38]. The Hungarian algorithm can be used to obtain a matching relationship with the minimum total distance, which is the distance between the two piece shapes.

In this study, matching experiments are performed on nine groups of common piece images for practical applications. Each group of pieces has 11 similar forms, and the groups are labeled A–I, as shown in Figure 6. The first piece of each group is matched with the following 22 pieces, including itself, and the first piece of the last group is matched with groups I and A, respectively. The value of the shape distance is shown in Figure 7. The distance between the first piece of each group and that between the first piece of the other 8 groups are calculated and matched, as shown in Figure 8. The results indicate that the distances between different pieces vary, with pieces of the same group having better matching results (smaller distance values). Naturally, when the piece image matches itself, the distance value is the minimum. It is observed that the highest similarity in results
is obtained for a distance value of 6.1. Therefore, we set 6.1 as the limit of the matching distance when matching the target and source tasks.

![Diagram](image)

**FIGURE 9.** Optimal sequence prediction process of target task pieces.

2) PREDICTION OPERATOR FOR OPTIMIZING THE PIECE SEQUENCE

The matching results of the pieces in the target and source tasks are stored in the set MC. Using the matching result MC to predict the order of the pieces in the target task set is integral to the transfer of the packing sequence. Each type of piece in the target task is matched with pieces in different cases in the ISM, where the historical task case with the minimum average distance is selected as the source piece set for the matching task.

When the distance value between the target task pieces and the pieces in the current source task is matched, the pieces of the same class will be clustered first, and the pieces of the source task with the minimum matching distance value are selected as the matching result. In addition, the source pieces that matched will be hidden, that is, pieces that have been successfully matched will be prevented from participating in the next match. When the distance value is greater than 6.1, it is considered that there is no matching piece in the source task set with the current target task pieces. The pieces of the source task that are successfully matched will be stored in the set MC to guide the sequencing of the matched pieces in the target task. Let the set of pieces of the target task be \( P_0 = \{P_{0,1}, P_{0,2}, \ldots, P_{0,n}\} \), and the pieces set corresponding to MC is \( P_s = \{P_{s,1}, P_{s,2}, \ldots, P_{s,m}\} \). The prediction process of target task piece sequencing is shown in Figure 9. The pieces in MC and \( P_0 \) The pieces in MC and \( P_0 \) are matched one by one, inheriting the order and angle of the pieces in set MC, so the order of pieces in \( P_0 \) is \( (P_{0,1}, P_{0,3}, P_{0,5}, P_{0,4}, P_{0,2}) \).

B. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart [47] proposed the theory of PSO by observing the predatory behavior of birds. The concept of the algorithm is relatively simple, as there are few parameters to be adjusted, and the convergence is fast. Here, the compression factor particle swarm algorithm with the adaptive function is used for optimization, and the transfer operation (based on piece matching) is used to determine the optimal position of each piece to improve the utilization rate of the plate. The process of the sequence transfer-based particle swarm optimization algorithm is presented in Figure 10.

**Algorithm 3** Particle Swarm Initialization Strategy Based on Sequence Transfer

1) **INITIALIATION**

In the PSO process, a particle represents a solution in the solution space, where the length of the particle is determined by the number of pieces. It should be noted that different sequences of packing the pieces can form different particles. For example, sorting according to the size of the area constitutes a particle, and changing the order of some pieces constitutes another particle. The target task set of particles is \( P_0 = \{P_{0,1}, P_{0,2}, \ldots, P_{0,m}\} \), where \( P_{0,i} \) is \((x_{0,i}, y_{0,i}, x_{02,i}, y_{02,i})\), representing the position coordinates of the bottommost leftmost and the lowest center of gravity of the piece \( i \) respectively. The vertex with the bottommost-leftmost and the lowest center of gravity is selected to evaluate the criteria proposed in formula (7), and the position with the best evaluation is determined as the placement location. According to Algorithm 1 and 2 and the optimal piece prediction operator, the optimal sequence of the packing of the historical pieces can be transferred to new packing tasks. Meanwhile, if the matching boundary (here, a value of 6.1) is exceeded, the matching will be considered unsuccessful, and the particles will be randomized. The piece sequences after transfer, and of the random pieces, are considered the initial particles. The specific steps of the proposed particle swarm initialization strategy are given in Algorithm 3.

2) **FITNESS FUNCTION AND PARTICLE SWARM UPDATE**

For each particle, that is, every optimized solution of all pieces to be packed, the fitness function value is calculated based on the fitness function. In this study, the fitness function is the objective function expressed in formula (1). Here, a high fitness function corresponds to a small packing height and high utilization rate. When calculating the fitness function, the adaptive adjustment strategy is applied to fine-tune the packing position of the pieces in the particles. The pieces should avoid overlapping and exceeding the plate edges according to the NFP strategy. After updating the speed and position of the particle, the position of the piece in the particle is not necessarily an integer due to the calculation method that allows non-integer numbers; hence, the rounding principle is used to convert them to integers. After the adjustment, the next round of evaluation is performed, and the
adaptive adjustment is performed again according to the NFP and positioning strategies until the criteria in formula (2) - (4) are satisfied. In the case of comparing the fitness function values of the front and back particles, if the packing mode of a piece in the current particle is different from that of the corresponding piece in the current optimal solution of PSO, the packing mode of the optimal solution is used, and the fitness function values are calculated before and after the conversion, where larger values represent the packing solution of the current particles.

When comparing the fitness function values of each particle with the individual extreme value $P_{\text{Best}}$, if the current value is higher than $P_{\text{Best}}$, the current value is set as the new $P_{\text{Best}}$ and the current position $x_{id}$ of the particle is set as the new corresponding position $X_{id}^p$. When comparing the fitness function values of all particles with the global extremum $Q_{\text{Best}}$, if the current value is higher than $Q_{\text{Best}}$, the current value is set as the new $Q_{\text{Best}}$ and the current position $x_{id}$ of particles is set as the new corresponding position $X_{id}^q$. The formulae for updating the velocity and position of the particle are expressed in formulas (16) and (17).

$$v_{id} = K \cdot [v_{id} + c_1 \cdot \text{rand}_1() \cdot (X_{id}^p - x_{id}) \,+ c_2 \cdot \text{rand}_2() \cdot (X_{id}^q - x_{id})] \tag{16}$$

$$x_{id} = x_{id} + v_{id} \tag{17}$$

Here $c_1$ and $c_2$ are learning factors; $\text{rand}_1()$ and $\text{rand}_2()$ are random numbers in the interval $[0,1]$, and $v_{id}$ and $x_{id}$ are the current velocity and position of the particles, respectively.

Here, $v_{id}$ is limited by the maximum velocity $v_{max}$. All values exceeding $v_{max}$ will be replaced by $v_{max}$. In addition, $K$ is the compression factor, as defined by formula (18).

$$K = \frac{2}{\sqrt{2 - \phi - \sqrt{\phi^2 - 4\phi}}} \tag{18}$$

Here, $\phi = c_1 + c_2$; $\phi > 4$; it usually takes $c_1 = c_2 = 2.05$, $K = 0.729$ in the compression factor PSO algorithm. In addition, related literature [48] shows that taking $v_{max}$ as the dynamic range of the particle can significantly improve the performance of the PSO algorithm. Therefore, when the plate length is unlimited and the width is fixed, the width range of the plate is typically used as the dynamic range of the particles.

3) STOP RULE

In general, an algorithm is stopped when it reaches the maximum number of iterations or when the optimal fitness function value does not change significantly in 30 consecutive steps. After the algorithm stops, the packing result with the optimal fitness function value is provided as the output, and the packing sequence and piece angle corresponding to the optimal packing result are returned to the ISM as a case of historical piece packing.

V. COMPUTATIONAL EXPERIMENTS

A. PROBLEM INSTANCES

The algorithm is written in Visual Studio C++. Computations tests were performed on a machine with a 2.9 GHz Intel I5-9400 CPU with 4 kernels and 8 GB of RAM. To test the performance of the proposed algorithm, actual data are sampled from a heavy industry company, and relevant data are extracted from the Special Interest Group on Cutting and Packing (SICUP, http://paginas.fe.up.pt/~esicup/). The sample information of these data is provided in Table 1. The parameters of the PSO that were incorporated into sequence transfer were set as follows: the limit of the similarity value is 6.1, the number of target task pieces is taken as the number of particles, the learning factor is 2.05, the compression factor is 0.729, the maximum velocity of the particle is the dynamic range of the particle (which is determined by the width of the plate), and the algorithm ended when the optimal fitness function value did not change significantly in 30 consecutive steps.

B. THE COMPUTATIONAL RESULTS ON THE PERFORMANCE OF THE ST-PSO

For each case, the ST-PSO algorithm is run 20 times. The results, including the utilization percentage and calculation time, obtained using a conventional algorithm or the ST-PSO algorithm are listed in Table 2 and Table 3, respectively. These data show that the ST-PSO algorithm has clear benefits in multiple case samples. Compared with the artificial packing sample before transfer, the ST-PSO algorithm proposed here can inherit and transfer the packing sequence of pieces in

![FIGURE 10. Process of the sequence transfer-based particle swarm optimization algorithm for the irregular packing problem.](image-url)
TABLE 1. Sample information.

| Instance | Number of pieces types | Total number of pieces | Sheet width | Rotational constraints |
|----------|------------------------|------------------------|-------------|------------------------|
| M8-1007  | 28                     | 91                     | 2000        | (0°, 90°, 180°, 270°)  |
| 200T-3054| 6                      | 36                     | 2000        | (0°, 90°, 180°, 270°)  |
| 200T-3041| 17                     | 38                     | 2000        | (0°, 90°, 180°, 270°)  |
| 200T-3010| 20                     | 102                    | 2000        | (0°, 90°, 180°, 270°)  |
| Dighel   | 16                     | 16                     | 100         | (0°)                   |
| Jakobs1  | 25                     | 25                     | 40          | (0°, 90°, 180°, 270°)  |
| Marques  | 8                      | 24                     | 104         | (0°, 90°, 180°, 270°)  |
| Shapes1  | 4                      | 43                     | 40          | (0°, 180°)             |

TABLE 2. Traditional algorithm result information.

| Instance | Artificial sample data before transfer | BLF | RKGA-NP |
|----------|----------------------------------------|-----|---------|
| M8-1007  | 86.6 480.16                           | --  | --      |
| 200T-3054| 76.1 360.52                           | --  | --      |
| 200T-3041| 82.4 450.66                           | --  | --      |
| 200T-3010| 76.5 390.56                           | --  | --      |
| Dighel   | 76.7 161.88                           | 77.40 8.87 | 100 99.96 | 2053 |
| Jakobs1  | 81.7 250.46                           | 82.60 43.49 | 81.67 78.79 | 10268 |
| Marques  | 83.2 241.80                           | 86.50 4.87  | 84.47 84.31 | 6871 |
| Shapes1  | 65.6 431.58                           | 66.50 2.19  | 66.37 65.98 | 3025 |

TABLE 3. Algorithm result information.

| Instance | Sequential random data | ST-PSO |
|----------|------------------------|--------|
| M8-1007  | 85.52 148.42           | 87.5 86.57 83.72 |
| 200T-3054| 76.28 85.9             | 76.3 75.48 29.3 |
| 200T-3041| 84.3 150.54            | 84.6 82.99 58.83 |
| 200T-3010| 73.36 89.66            | 73.8 71.57 52   |
| Dighel   | 74.1 21.24             | 73.6 72.44 5.44 |
| Jakobs1  | 80.23 72.16            | 80.5 79.65 38.76 |
| Marques  | 83.6 12.15             | 84.4 83.67 3.90 |
| Shapes1  | 65.44 6.88             | 65.9 65.24 3.61 |

The original samples, and also calculate the packing positions reasonable well. As a result, five better results of the ST-PSO (M8-1007, 200T-3054, 200T-3041, Marques, Shapes1) can be observed. There are two samples (Marques, Shapes1) that can be compared with the BLF [49] or RKGA-N [50] algorithms, and the utilization deviation is within 1%. Furthermore, in the case of a random sequence and using the same position, the ST-PSO algorithm achieves similar or better utilization results in a shorter time for some cases. This is similar to the comparative change rule of sample data before transfer, indicating the effectiveness of sequence transfer. It is worth noting that in almost all cases, the ST-PSO algorithm shows a higher calculation speed than other methods, and changes with the quality of the sample data. The layouts for the corresponding results are shown in the Appendix.

C. DISCUSSION

According to the experimental results, the ST-PSO algorithm has a better optimization effect than the packing data before transfer, and its performance is comparable to that of some heuristic algorithm. The results are analyzed considering case samples, and the mechanism and structure of the ST-PSO algorithm. The case samples consist of the actual production data obtained by a heavy industry company and some data processed by the company’s artificial experience, which have a more complex piece structure. According to the actual production process, 0°, 90°, 180°, and 270° angles are selected. Orthogonal rotations of the pieces can effectively reduce the search space. Since the case samples are mainly generated by the conventional heuristic algorithm combined with artificial intervention, and the ST-PSO algorithm can transfer an optimal packing sequence of the original samples for the initialization of the particle swarm, the ST-PSO algorithm can reduce computing time. However, the samples before the transfer of artificial intervention can pack small pieces within larger pieces based on artificial experience, which can increase the utilization rate during the packing of some pieces. This is also an advantage of artificial experience packing. However, the packing of large numbers of pieces and pieces without nesting appears to be inadequate. Considering the mechanism of the ST-PSO algorithm, it can obtain a good balance between global exploration and local refinement. The algorithm has few parameters that need to be adjusted and fast convergence. In addition, it can simultaneously use both individual local information and group global information to guide the search, combined with the initial particle swarm, to obtain a better optimal solution than other algorithms. The ST-PSO algorithm has a modified process and structure compared to the conventional PSO algorithm, which requires that the speed and position of the particles are initialized randomly. Further, in the conventional PSO algorithm, the initial fitness value of each particle is evaluated and the current optimal initial fitness value is taken as the local optimum, which results in a high computational time and the risk of convergence to a local optimum. However, in the ST-PSO algorithm, the initial particles are transferred from the sequence of excellent cases, instead of performing a random search from the initial state, which makes it easier to obtain the global optimal solution as the probability of falling into local optimaums is reduced. As shown in Algorithm 3 and Table 3, the particle swarm is initialized by the transfer method, which not only prevents the search of particles from falling into the local optimum, but also shortens the search time of the algorithm, which contributes a lot to the saving of time cost. In addition, an improved positioning strategy based on evaluation criteria is used to solve the packing problem.
problem in the ST-PSO algorithm, which can better guide the calculation of particle fitness to update the particle swarm information and parameters. Compared with the conventional BL algorithm, the positioning algorithm of hybrid BL has a higher complexity, which has a better effect on solving the problem of pieces stacking on both sides and the unused hole area of the plate. Therefore, the algorithm has a positive contribution to the placement searching of pieces packing.

Although the ST-PSO algorithm can successfully solve the problem of 2D irregular pieces packing, it also has some limitations. First, the algorithm relies on the existing excellent case samples to obtain a better packing sequence, and then guide the initialization of the particle swarm. Second, the rotation angle of the pieces does not have the same freedom as conventional artificial experience packing, and the overall search range is limited, which depends on many factors, including the acceptable search time, algorithm design, industrial application, and research conclusions of related literature [51]. Third, although the sequencing strategy in ST-PSO algorithm can save some time cost, the positioning algorithm is more complicated and time-consuming than the conventional BL algorithm because of its stronger search ability. Nonetheless, compared with the case samples before transfer, the ST-PSO algorithm can achieve better packing results while inheriting the excellent packing sequence.

VI. CONCLUSION
A novel fastPSO algorithm incorporating sequence transfer is developed for 2D irregular piece packing optimization. The packing sequence of the historically optimal solution is transferred, and the positioning strategy is improved using the evaluation criteria. To exploit the historical information of the packing problem obtained from the transfer to increase the search efficiency of the algorithm, a piece matching method based on the shape context similarity is used, where the particle swarm is initialized using the prediction operator of the piece sequence. Finally, the result of optimal packing is used to update the ISM.

The experimental results show that the ST-PSO algorithm has good robustness. It can inherit and transfer the historical packing sequence in a short time, while retaining or even exceeding the typical packing data in the industry to ensure that it has strong practical applicability. In the case of good historical data samples, the results are comparable to those of available heuristic algorithms. Moreover, the algorithm predicts a relatively short packing time, and the effect will be more obvious if better historical samples are used. Thus, good packing performance can be retained, while the time and cost of packing can be effectively reduced.

Although the ST-PSO algorithm achieved acceptable performance for the industry case study, the convergence speed and effect of the packing are limited when the packing sequence of the case sample is not accurate or the original utilization rate of piece packing is not high. In future work, we will collect more cases and samples to test the performance of ST-PSO. In addition, the algorithm will be extended to include deep learning and global transfer learning in an attempt to solve the packing problem of batch production with multi-constraint variables.

APPENDIX
REFERENCES

[1] Y. N. Song, “Study of algorithms of graphic-matching, control of packing and contacting in two-dimensional packing system,” Ph.D. dissertation, Control Theory Control Eng., South China Univ. Technol., Guangzhou, China, 2004.

[2] Y. Li, “Approximate algorithm applied in stock cutting problems,” M.S. thesis, Huzhong Univ. Sci. Technol., Wuhan, China, 2005.

[3] E. P. Hoffman, M. R. Garey, D. S. Johnson, and R. E. Tarjan, “Performance bounds for level-oriented two-dimensional packing algorithms,” SIAM J. Comput., vol. 9, no. 4, pp. 808–826, Nov. 1980, doi: 10.1137/0209090.

[4] Chazelle, “The bottom-left bin-packing heuristic: An efficient implementation,” IEEE Trans. Comput., vol. C-32, no. 8, pp. 697–707, Aug. 1983.

[5] J.-F. Tsai and H.-L. Li, “A global optimization method for packing problems,” Eng. Optim., vol. 38, no. 6, pp. 687–700, Sep. 2006.

[6] P. K. Agrawal, “Minimising trim loss in cutting rectangular blanks of a single size from a rectangular sheet using orthogonal guillotine cuts,” Eur. J. Oper. Res., vol. 64, no. 3, pp. 410–422, Feb. 1993.

[7] E. López-Camacho, G. Ochoa, H. Terashima-Marin, and E. K. Burke, “An effective heuristic for the two-dimensional irregular bin packing problem,” Ann. Oper. Res., vol. 206, no. 1, pp. 241–264, Jul. 2013.

[8] A. K. Sato, T. C. de Martins, and M. S. G. de Tsuzuki, “A pairwise exact placement algorithm for the irregular nesting problem,” Int. J. Comput. Integ. Manuf., vol. 29, no. 11, pp. 1177–1189, Nov. 2016.

[9] J. F. Oliveira, A. M. Gomes, and J. S. Ferreira, “TOPOS-A new constructive algorithm for nesting problems,” OR-Spektrum, vol. 22, no. 2, pp. 263–284, 2000.

[10] H. S. Ismail and K. K. B. Hon, “New approaches for the nesting of two-dimensional shapes for press tool design,” Int. J. Prod. Res., vol. 30, no. 4, pp. 825–837, Apr. 1992.

[11] R. Heckmann and T. Lengauer, “Computing closely matching upper and lower bounds on textile nesting problems,” Eur. J. Oper. Res., vol. 108, no. 3, pp. 473–489, Aug. 1998.

[12] M. K. Omar and K. Ramakrishnan, “Solving non-oriented two-dimensional bin packing problem using evolutionary particle swarm optimisation,” Int. J. Prod. Res., vol. 51, no. 20, pp. 6002–6016, Oct. 2013.

[13] H. Dong and J. Chen, “Research and application of an improved particle swarm algorithm in two-dimensional layout,” J. Zhejiang Univ. Technol., vol. 44, no. 4, pp. 388–391, 2016.

[14] A. R. Babu and N. R. Babu, “A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms,” Comput.-Aided Des., vol. 33, no. 12, pp. 879–891, 2001, doi: 10.1016/S0010-4485(00)00112-3.

[15] J. A. Bennell and K. A. Dowsland, “A Tabu thresholding implementation for the irregular stock cutting problem,” Int. J. Prod. Res., vol. 37, no. 18, pp. 4259–4275, Dec. 1999.

[16] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 26, no. 1, pp. 29–41, Feb. 1996.

[17] J. Levine and F. Ducatelle, “Ant colony optimization and local search for layout problems,” M.S. thesis, Huazhong Univ. Sci. Technol., Wuhan, China, 2004.

[18] J. F. Oliveira, A. M. Gomes, and J. S. Ferreira, “TOPOS-A new constructive algorithm for nesting problems,” OR-Spektrum, vol. 22, no. 2, pp. 263–284, 2000.

[19] Y. N. Song, “Study of algorithms of graphic-matching, control of packing and contacting in two-dimensional packing system,” Ph.D. dissertation, Control Theory Control Eng., South China Univ. Technol., Guangzhou, China, 2004.

[20] M. S. thesis, China Univ. Mining Technol., Xuzhou, China, 2019.

[21] X.-S. Wang, J. Pan, and Y.-H. Cheng, “Ant-Q algorithm based on knowledge transfer,” Dianzhi Xuebao Acta Electronica Sinica, vol. 39, no. 10, pp. 2359–2365, 2011.

[22] X.-S. Wang, J. Pan, and Y.-H. Cheng, “Ant-Q algorithm based on knowledge transfer,” Dianzhi Xuebao Acta Electronica Sinica, vol. 39, no. 10, pp. 2359–2365, 2011.
[51] D. Dewei and Y. Yunhui, “Optimal packing of irregular parts on plates with defects,” J. Graph., vol. 34, no. 2, pp. 31–37, 2013, doi: 10.3969/j.issn.2095-302X.2013.02.006.

JIE FANG received the master’s degree in mechanical engineering from Central South University, Changsha, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. His research interests include intelligent manufacturing, computational intelligence, and machine learning for combinatorial optimization.

YUNQING RAO was born in Chibi, Hubei, China, in 1968. He received the M.S. degree in mechanical manufacturing and the Ph.D. degree in mechanical manufacturing and automation from Huazhong University of Science and Technology, Wuhan, China, in 1992 and 1999, respectively.

In 2001, he was with the Management of Industrial Production (MIP) Training, Aachen University of Technology, supported by Chiang’s Fund of Hong Kong. From 2004 to 2005, he visited the University of Oxford. He is currently a Professor with Huazhong University of Science and Technology. He has authored over 200 journal articles. His research interests include operation optimization of manufacturing systems, manufacturing execution systems (MES), and intelligent manufacturing.

PAN LIU received the B.S. degree in industrial engineering from Huazhong University of Science and Technology, Wuhan, China, in 2018, where he is currently pursuing the M.S. degree in mechanical engineering. His current research interests include intelligent manufacturing, intelligent optimization algorithms, and computer vision.

XUSHENG ZHAO received the B.S. degree in mechanical engineering from Hubei University of Technology, in 2015, and the M.S. degree in renewable energy engineering from Huazhong University of Science and Technology, in 2019, where he is currently pursuing the Ph.D. degree with the School of Mechanical Science and Engineering. His research interests include adaptive machining, multi-axis inspection technology, and machine learning for combinatorial optimization.