



OPEN A new approach for bin packing problem using knowledge reuse and improved heuristic

Jie Fang^{1,2,3}, Xubing Chen^{1,2}✉, Yunqing Rao⁴, Yili Peng^{1,2} & kuan Yan²

The two-dimensional (2D) irregular packing problem is a combinatorial optimization problem with NP-complete characteristics, which is common in the production process of clothing, ships, and plate metals. The classic packing solution is a hybrid algorithm based on heuristic positioning and meta-heuristic sequencing, which has the problems of complex solving rules and high time cost. In this study, the similarity measurement method based on the twin neural network model is used to evaluate the similarity of pieces in the source task and the target task. The reusability evaluation of packing tasks is designed to select appropriate source task knowledge. The transfer operator is used to transfer the piece sequence knowledge from the source task to complete the reuse of packing knowledge in the target task. The bottom-left algorithm is improved to complete the placement of 2D irregular pieces. The computational experiments show that the proposed algorithm for the bin packing problem using knowledge reuse and improved heuristic (KRIH) has good robustness. The KRIH algorithm can obtain 8 equal or better results on 16 instances in a relatively short time compared with some classical heuristic algorithms, which has good application potential.

Keywords Packing problem, Irregular piece, Knowledge reuse, Knowledge transfer, Heuristic algorithms

Packing problems are commonly found in industries such as mechanical manufacturing, aviation, clothing, automotive manufacturing, shipbuilding, and furniture manufacturing. The manufacturing process of typical heavy industry products is shown in Fig. 1. In actual production, efficient packing algorithms can improve the utilization of raw materials and reduce production costs within an acceptable time. In a sense, the reduction of raw material consumption can have a beneficial impact on the environment¹. Packing problems, also known as bin packing problems or nesting problems, can be divided into three categories²: the one-dimensional (1D) packing problem (wire cutting stock problem)^{3,4}, the two-dimensional (2D) packing problem (planar packing problem)^{5,6} and the three-dimensional (3D) packing problem⁷. The 2D packing problem is an interesting combinatorial optimization problem, which is used to place pieces of different shapes on one or more plates according to certain rules to minimize the space occupied by pieces or maximize the utilization of raw materials. The pieces here are general terms. In the metal processing industry, it refers to the pieces to be processed. In the garment manufacturing industry, it refers to the sample to be cut. And in the logistics industry, it refers to the goods to be placed. The packing problem has a large solution space and is an NP-complete (NP-C) problem, and the optimal solution cannot be obtained in polynomial time⁸. According to the regularity of piece shape, 2D packing problems can be further divided into 2D regular and 2D irregular packing problems. Among them, 2D irregular packing problems have received widespread attention due to their more diverse piece shapes and complex packing operations. Compared with the rectangular packing problem, the 2D irregular piece packing problem is very different in constraint calculation, solution strategy, overlap detection, so it has a larger solution space and more complicated technical operation. The 2D irregular piece packing has more applications in actual production. For 2D irregular packing problems, the existing research focuses on finding the approximate optimal solution or satisfactory solution within an acceptable time. The solving of the 2D irregular packing problem often involves two stages: sequence optimization and position optimization. In the sequence optimization stage, the sequence of pieces arranged on the plate is optimized. In the position optimization stage, the final placement scheme of pieces on the plate is determined. Early solutions to the irregular packing problem include

¹Hubei Provincial Key Laboratory of Chemical Equipment Intensification and Intrinsic Safety, Wuhan Institute of Technology, Wuhan 430205, P. R. China. ²School of Mechanical & Electrical Engineering, Wuhan Institute of Technology, Wuhan 430205, P. R. China. ³Hubei Engineering Research Center for Intelligent Detection and Identification of Complex Parts, Wuhan 430205, P. R. China. ⁴State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430205, P. R. China. ✉email: chenxb@wit.edu.cn

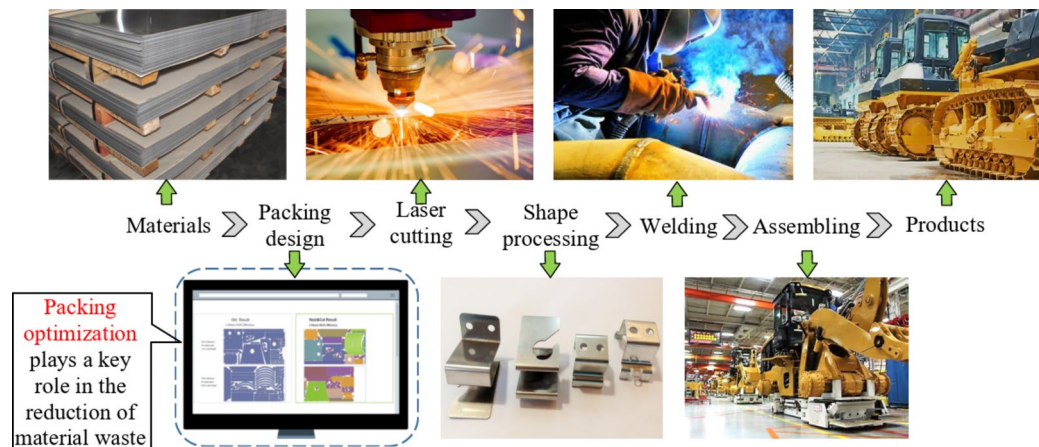


Fig. 1. The manufacturing process of typical heavy industry products.

the polygonal envelope method⁹, clustering feature combination method, and fitting method¹⁰, etc. In the late 1970s, researchers began to study contour features based on the real shape of irregular pieces to judge the overlap between pieces. These methods include no-fit polygon (NFP)^{11,12}, raster methods (also called pixel methods)¹³, linear programming (LP), and mixed-integer linear programming (MIP)^{14,15}. Each method has its advantages, but the current mainstream method is NFP¹⁶. Burke E et al.¹⁷ designed an NFP algorithm capable of dealing with circular arcs, improved the track sliding method, and achieved good test results.

In the process of piece placement, commonly used positioning algorithms include the bottom-left (BL) algorithm¹⁸, the improving bottom-left (iBL) algorithm¹⁹, etc. Utilizing and improving the BL algorithm becomes one of the research points for solving 2D irregular packing problems. The sequence of piece placement is crucial for the final packing result. Therefore, the study of sequence optimization of the packing problem is another key point to improve the piece packing efficiency. Sequence optimization algorithm belongs to the meta-heuristic algorithm^{20,21} or intelligent algorithm, including simulated annealing algorithm²², tabu search algorithm²³, genetic algorithm²⁴, particle swarm optimization algorithm²⁵, and colony algorithm²⁶, etc. Currently, the hybrid algorithm combining the heuristic algorithm and the meta-heuristic algorithm is the mainstream method for solving 2D irregular packing problems²⁷. Although the existing research on 2D irregular packing problems has made significant achievements, there are still some problems. On the one hand, the universality of the current algorithm is not high, and the performance of the algorithm on different instances is significantly different. On the other hand, current algorithms start searching from the zero states of the problem, which takes a long time and has weak knowledge mining ability, limiting the improvement of solving efficiency.

Real-world problems rarely exist in isolation. Therefore, when faced with unprecedented challenges or tasks, humans often turn to various information sources, including knowledge extracted from past problem-solving experiences²⁸. With the development of artificial intelligence technology, transfer learning technology, which uses data from related source tasks to help solve problems in new tasks or target tasks, has attracted wide attention. This technology has achieved great success in fields such as emotion classification, natural language processing, and computer vision^{29–31}. The relevant research achievement is largely limited to the field of predictive analysis. Although some scholars have explored combinatorial optimization problems using machine learning methods, there are few studies that apply transfer techniques to combinatorial optimization problems³². Currently, some scholars have attempted to use transfer optimization theory to solve some combinatorial optimization problems. Transfer optimization can be divided into sequential transfer optimization, multi-task optimization, and multi-form optimization according to different application methods³³. Sequential transfer optimization is a one-way knowledge transfer³⁴. Feng L et al.³⁵ proposed a memetic computational paradigm based on evolutionary optimization and transfer learning for searching, which enhanced the search ability for routing optimization problems. In the study of Abhishek Gupta et al.^{36,37}, transfer optimization was used to solve the black-box optimization problem that cumulatively encompass a range of ways in which transfer optimization can be put to use in practical settings. In the research on transfer optimization, some algorithms are mainly discussed, such as transfer Bayesian optimization, genetic transfer, evolutionary multitasking, and evolutionary algorithms with the model-based transfer³⁸. The study of D. O'Neill et al.³⁹ showed that the transfer of optimized genetic material from source to target symbolic regression task could help to reduce training errors and improve generalization performance. Ali Yaddaden et al.⁴⁰ adapted a state-of-the-art NCO model for implementing transfer learning. The experimental results of this study suggested that transfer learning may accelerate the training process while improving sample efficiency. In general, the most advanced algorithms used to solve combinatorial optimization problems rely on handcrafted heuristics for making decisions which have high calculation costs, and the machine learning method looks like a natural candidate to solve such problems in a more principled and optimized way. Although evolutionary transfer optimization has achieved great success in certain fields, they often overly rely on the sieving effect of evolutionary selection pressure, and the sufficiently in-depth analysis is seldom done to infer the similarity⁴¹ across problems. Moreover, most evolutionary computing methods still start the search from zero state⁴², resulting in low efficiency in problem solving. In recent years, some researchers have tried to

combine the heuristic algorithm with the transfer algorithm to solve the bin packing problem. Darius Scheepers et al.⁴³ explored the application of transfer learning to the one-dimensional box packing problem (1BPP), which achieved good results by transferring the best elements from each generation of the source hyper-heuristic to the target hyper-heuristic. Mohammad Alodat⁴⁴ used deep transfer learning (DTL) to facilitate the recognition of plant items products packing service. The DTL is used by training and improving from the pre-training network on the ImageNet of the convolutional neural network architecture. Fang J et al.⁴⁵ used the shape context algorithm to match the pieces in the target task with the pieces in the source task and transferred the optimal packing sequence of the source task to test the instances, and certain results were achieved. Xu Xiaofei et al.⁴⁶ combined the ant colony algorithm with the Q-learning algorithm and introduced dual-source linear knowledge transfer technology to solve the rectangular layout optimization problem. Guo B et al.¹⁶ showed in their recent research on irregular packing that machine learning algorithms may be effective in optimizing the packing sequence and then obtaining a good packing scheme, but there is a lack of exploration on the research direction. Overall, although there are few researches on solving packing problems using transfer learning, more and more attention is being paid. At present, the existing knowledge transfer studies on packing problems mainly focus on the transfer mode without considering the use of historical excellent layout diagrams for knowledge reuse, which limits the reference and improvement of excellent cases. Limited to the sample of the source task, some existing studies have only been tested on a limited number of instances. Therefore, the research on packing technology based on knowledge transfer not only promotes and inherits the existing research achievements but also provides a new idea for solving the packing problem, which has great theoretical significance and application potential.

In this paper, a new approach for the plate packing problem using knowledge reuse and improved heuristic (KRIH) is proposed based on the optimization of piece packing in heavy industry. Specifically, the piece similarity measurement method based on the twin neural network model is designed to evaluate the similarity of pieces in the source task and the target task. The reusability evaluation of the packing task is designed to select the appropriate source task knowledge. The transfer operator is used to transfer the pieces sequence from the source task to complete the reuse of the packing knowledge in the target task. To improve the efficiency of packing, the BL algorithm is improved to complete the placement of 2D irregular pieces. In addition, reinforcement learning based on Monte-Carlo is applied to improve the search of the packing sequence.

The 2D irregular packing problem is summarized firstly in this paper. Then, the mathematical modeling of 2D irregular pieces packing is described. Secondly, a hybrid position strategy based on fit and minimum envelope rectangle increment is explained. Furthermore, the shape similarity measurement method based on twin neural networks and the reusability evaluation of packing tasks based on pieces shape similarity is introduced. Thirdly, the transfer operator and reinforcement learning search used for the generation of the packing sequence are described. Finally, experimental parameters and experimental results are analyzed.

Mathematical model of packing problem

Due to different processing conditions and production technology in different fields, there are different mathematical models for packing optimization problems^{47,48}. Leao A et al.⁴⁹ took the 2D irregular nesting problem as the research object and summarized the establishment of the mathematical model. The mathematical model provides guidance for the determination of the final pieces packing scheme. According to the classification of Wäscher G² et al., the research object of this paper is 2D irregular pieces with a single and sufficient raw material. The plate is the 2D carrier for pieces packing, with a fixed width and unlimited length. The optimization goal of 2D irregular pieces packing is to obtain the minimum packing height or the maximum raw material utilization rate while meeting constraint conditions. Furthermore, the basic constraint of the packing problem is that pieces do not overlap each other, and all pieces do not exceed the boundaries of the plate. The objective function of 2D irregular pieces packing is shown in Formula (1), and the constraint conditions are shown in Formula (2).

$$\rho = \max_{i=1}^n S_i / WH \quad (1)$$

$$s.t. \begin{cases} P_i \subset Q \\ P_i \cap P_j = \emptyset \\ Q = \{p | 0 \leq p_y \leq W \cap p_x \geq 0\} \\ i \neq j, i, j \in 1, 2, \dots, n \end{cases} \quad (2)$$

Among them, ρ is the utilization rate of the plate, and the greater the ρ value, the better the optimization effect. W and H are the width and height of the plate, respectively, and S_i is the area of the pieces to be arranged. P_i and P_j represent the i -th and j -th pieces, respectively, which can be represented by the vertex p of the polygon formed by its outer contour. Q represents the region of the plate, p_x and p_y are the horizontal and vertical coordinates of the piece vertex, respectively.

Hybrid position strategy based on fit and minimum envelope rectangle increment No-fit polygon and inner no-fit polygon

No-fit polygon (NFP) is a real shape algorithm that can effectively calculate the overlap between pieces, which was first proposed by Art Jr RC⁵⁰ in 1966. The NFP method obtains the optimal position of the piece by finding all the relative positions of two pieces shape in different postures, which provides an accurate solution for piece placement. Although the definition of NFP is simple, its calculation is complex. There are four common ways to generate NFP: moving collision algorithm⁵¹, trajectory algorithm⁵², convex decomposition of polygons⁵³, and

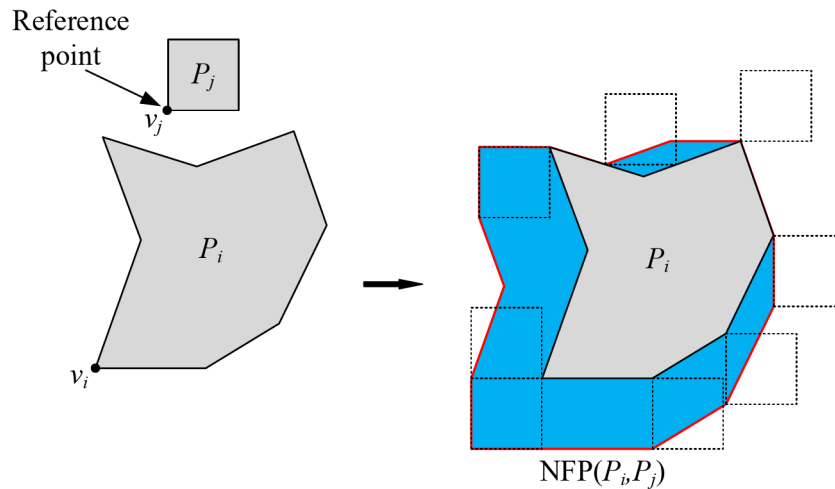


Fig. 2. Schematic diagram of the no-fit polygon $NFP(P_i, P_j)$.

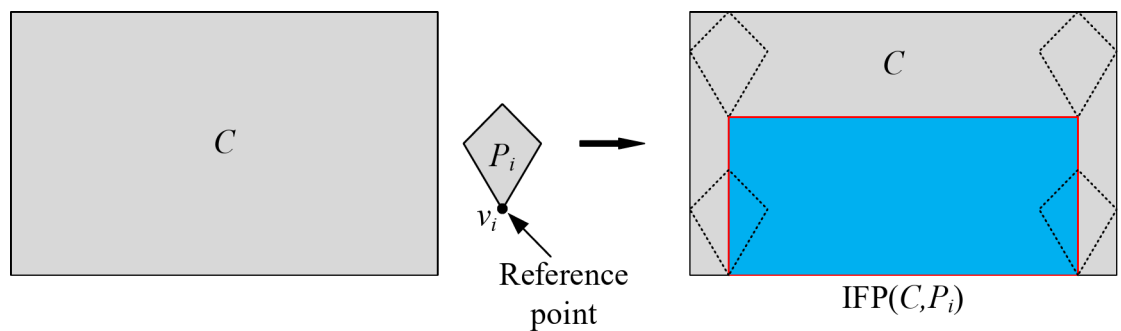


Fig. 3. Schematic diagram of the inner no-fit polygon $IFP(C, P_i)$.

Minkowski vector sum algorithm^{54,55}. The simple definition of NFP is as follows: Given two irregular pieces P_i and P_j , fix the piece P_i and let the piece P_j do a non-rotating translation motion around the piece P_i . During this process, keeping piece P_i and P_j in contact without overlapping. The closed polygon generated by the fixed reference point v_i on the piece P_i is the no-fit polygon of the piece P_j relative to the piece P_i , denoted as $NFP(P_i, P_j)$. When the reference point v_j is inside the $NFP(P_i, P_j)$, the piece P_i and P_j will overlap; When the reference point v_j is outside the $NFP(P_i, P_j)$, the piece P_i and P_j do not overlap, and there is a gap; When the reference point v_j is on the boundary of $NFP(P_i, P_j)$, the piece P_i and P_j will contact with each other, which is also the most reasonable state. The generation process of $NFP(P_i, P_j)$ is shown in Fig. 2.

Similar to the NFP, the inner no-fit polygon (IFP) is a tool to calculate the boundary overlap relationship between pieces and the plate. Assuming that the size of the plate is a certain value that is much larger than the size of the piece, the simple definition of IFP is as follows: given an irregular piece P_i and a rectangular plate C , let the piece P_i move in a non-rotating translation motion inside the plate C . During this process, the boundary of plate C and piece P_i is kept just in contact. The smaller rectangle generated by any fixed reference point v_i on piece P_i is the inner no-fit polygon of piece P_i relative to plate C , which is denoted as $IFP(C, P_i)$. When the reference point v_i is inside the $IFP(C, P_i)$, piece P_i is inside the plate C ; When the reference point v_i is outside the $IFP(C, P_i)$, piece P_i will exceed the plate, which does not meet the constraint condition; When the reference point v_i is exactly on the boundary of $IFP(C, P_i)$, the piece P_i is will contact with the plate C , which is the most reasonable state. The generation process of $IFP(C, P_i)$ is shown in Fig. 3.

According to the mathematical constraint of irregular pieces packing, the corresponding reference points on the piece should not only fall on the outside or boundary of the NFP but also on the inside or boundary of the IFP. Therefore, the final feasible packing area is denoted as the final no-fit polygon $FNFP(P_i)$, which can be obtained by Formula (3). P_i represents the i -th piece to be packed and P_j represents the j -th piece to be packed.

$$FNFP(P_i) = IFP(C, P_i) - NFP\left(P_i, \sum_{j=1}^{i-1} P_j\right) \quad (3)$$

Improved hybrid position algorithm

The BL algorithm^{17,56} is one of the most popular position algorithms due to its simple rules and parameter settings. However, the BL algorithm only considers placing the pieces as far to the left as possible and only takes shortening the occupied length of the plate as the optimization goal, ignoring the global situation and the special fit relationship between pieces. On the basis of the BL algorithm, a hybrid position strategy based on fit and minimum envelope rectangle increment (HPS) is proposed in the paper. The pseudo-code of HPS is shown in Algorithm 1. Compared with the traditional BL algorithm, HPS algorithm takes into account the global situation and the special mating relationship between pieces, which can reduce the generation of plate holes. The improved positioning strategy is to select a local optimal reference point standard on the boundary of FNFP so that the pieces can be arranged more reasonably on the plate. Given a piece P , the area of the envelope rectangle of the irregular piece P is defined as S_{SP} and the overlapping area of the envelope rectangle of the i -th piece is defined as S_{Si} , then the fit degree M of the piece P is shown in Formula (4), where k is the number of pieces that have already been packed.

$$M = \sum_{i=1}^k S_{Si} / S_{SP} \quad (4)$$

Using the concept of a simple envelope rectangle, an envelope rectangle composed of multiple pieces can be defined, which is composed of the maximum and minimum horizontal and vertical coordinates in the vertices of all pieces. The area of the envelope rectangle before the piece is packed is defined as S_1 , and the area of the new envelope rectangle formed after the piece is packed is defined as S_2 . The increment of the envelope rectangle area \hat{i} can be obtained by the Formula (5). The HPS algorithm can be described as follows: the threshold of fit degree M is set to 1. When $M \geq 1$, the algorithm can mark this candidate point and enter the marked point search. When $M < 1$, the left point on the rectangular piece is selected. When multiple points are to the left at the same time, the point with the smallest area increment \hat{i} is selected. The lowest point is selected when the area increment is also the same.

$$\hat{i} = S_2 - S_1 \quad (5)$$

$pv=(+\infty, +\infty)$, isAmnesty=False, $\min\hat{i}=+\infty$

for v in FNFP:

calculate M, \hat{i}

if isAmnesty:

if $M \geq 1$:

if $(v.x < pv.x)$ or $(v.x == pv.x \text{ and } \hat{i} < \min\hat{i})$ or $(v.x == pv.x \text{ and } \hat{i} == \min\hat{i} \text{ and } v.y < pv.y)$:

$pv = v, \min\hat{i} = \hat{i}$

else:

if $M \geq 1$:

isAmnesty=True, $pv = v, \min\hat{i} = \hat{i}$

elif $(v.x < pv.x)$ or $(v.x == pv.x \text{ and } \hat{i} < \min\hat{i})$ or $(v.x == pv.x \text{ and } \hat{i} == \min\hat{i} \text{ and } v.y < pv.y)$:

$pv = v, \min\hat{i} = \hat{i}$

return pv

Algorithm 1. HPS algorithm

Pieces matching and packing tasks reusability evaluation

Shape similarity measurement method based on twin neural network

With the continuous development and updates of machine learning technology, the development of deep learning technology represented by the convolutional neural network³⁷ tends to mature. In this paper, a twin neural network structure with the convolutional neural network as the core is designed to measure the similarity of 2D pieces shape and the advantage of the convolutional neural network in feature extraction is used. Twin neural networks were first proposed by Bromley et al.⁵⁸ in 1993 to compare check signatures of banks. Compared with traditional algorithms such as shape context, the method based on twin neural networks can utilize convolutional neural networks to extract shape features, avoiding the complexity and error of manual design.

Assuming that the two input piece graphics to be compared are A_1 and A_2 , respectively. After the feature extraction network G_C , the final feature vectors obtained are $G_C(A_1)$ and $G_C(A_2)$, respectively. E_C is the similarity measurement network. The algorithm of piece similarity measurement based on twin neural networks is divided into three steps: graph rotation and direction correction, shape feature extraction, and similarity measurement.

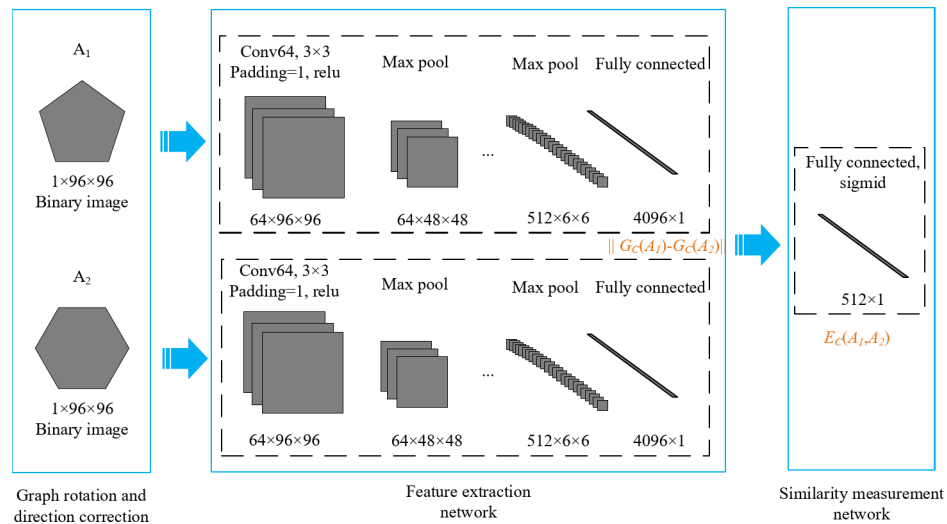


Fig. 4. Diagram of shape similarity measurement algorithm.

The algorithm structure of the shape similarity measurement based on the twin neural network is shown in Fig. 4.

Correction of graphic rotation direction

The feature extraction network proposed in this paper adopts the convolutional neural network, which has weak rotation invariance. Before the pieces are matched, the graphics need to be preprocessed, and the rotation angle of the graphics needs to be corrected. Each 2D image has a main axis that rotates with the graphics. After the direction angle of the main axis is calculated, the graphics can be uniformly rotated to a state where the main axis is horizontal or vertical, and the rotation direction of the graphics can be automatically corrected. The shapes used for training in this paper adopt the binary image format, with the size of 96×96 pixels. The main axis direction is obtained by calculating the central moment. As the piece shape is filled with a single color, the piece belongs to the single channel graphics. Set the grayscale image of a single channel of the piece to $f(x, y)$, p and q are contour collection points on the piece shape. The geometric moments of order $p+q$ are calculated as shown in Formula (6). After obtaining the zero-order origin moment and first-order origin moment of the piece shape, the center of gravity coordinates of the piece shape can be calculated as shown in Formula (7). For single channel grayscale image $f(x, y)$, the center distance M_{pq} and normalized center distance μ_{pq} of the $p+q$ order can be calculated respectively according to the center of gravity coordinates of the piece shape, as shown in Formulas (8) and (9). Setting θ as the main axis direction angle of the piece graphics, which can be calculated according to Formula (10) and (11).

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (6)$$

$$x_c = m_{10}/m_{00}, \quad y_c = m_{01}/m_{00} \quad (7)$$

$$M_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x, y) \quad (8)$$

$$\mu_{pq} = M_{pq} / M_{00}^{\frac{p+q}{2}+1} \quad (9)$$

$$\tan 2\theta = 2\mu_{11} / \mu_{20} - \mu_{02} = 2M_{11} / M_{20} - M_{02} \quad (10)$$

$$\theta = \frac{1}{2} \text{atan2}(2M_{11}, M_{20} - M_{02}) \quad (11)$$

Where m_{pq} is the geometric moment of order $p+q$, and (x, y) is the pixel coordinate. By utilizing the main axis direction angle θ , the piece graphics can be rotated, and the main axis of the graph can be returned to a fixed direction, which makes the piece graphics have good rotation invariance.

Feature extraction network

The VGG16 deep convolutional neural network proposed by Simonyan K et al.⁵⁹ shows excellent performance in image recognition and classification tasks. In this paper, based on the VGG16 deep convolutional neural network, the image size and channel number are improved to adapt to the feature extraction of 2D pieces graphics. The improved feature extraction network is shown in Fig. 5. After being preprocessed by main axis rotation and size

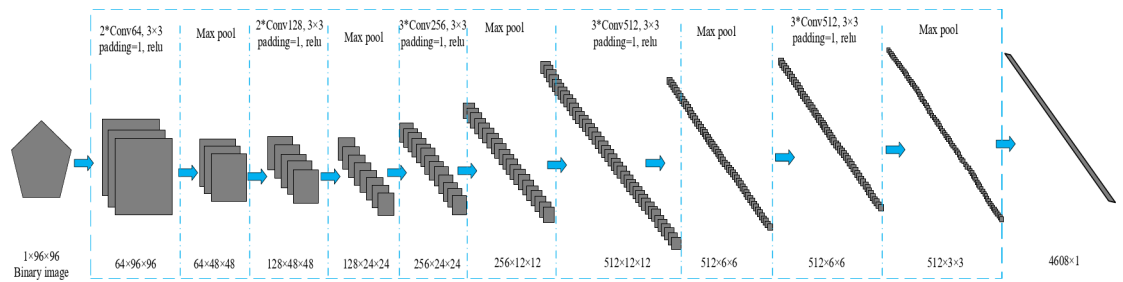


Fig. 5. Schematic diagram of feature extraction network processing.

scaling, the input shape is transformed into a binary image with a size of 96×96 pixels. The 1D feature vector with a length of 4608 is output after the feature extraction network processing.

Similarity measurement network

After inputting the piece graphics A_1 and A_2 to be compared, feature vectors $G_C(A_1)$ and $G_C(A_2)$ of 4608×1 are finally output through the feature extraction network GC processing. The similarity measurement vector H of piece graphics A_1 and A_2 is calculated as shown in Formula (12). The use of absolute value can avoid errors caused by the different input sequence of piece graphics A_1 and A_2 . The two-layer fully connected network is used to judge the similarity of the graphics. Meanwhile, the sigmoid activation function is used to process the final output data so that the value range of the result is mapped between 0 and 1 and is used as the similarity of piece graphics A_1 and A_2 .

$$H_i = |G_C(A_1)_i - G_C(A_2)_i| \quad (12)$$

Acquisition and identification of pieces in the layout diagram

As the case source of knowledge transfer, the acquisition and application of source tasks are crucial for the transfer result. The source task of this paper comes from the existing excellent layout cases, including enterprise cases and layout diagrams obtained by excellent algorithms. Through the analysis and application of excellent layout diagrams, the target tasks which adopt deformation design can be quickly guided and packed, thus improving the packing efficiency and fitness of target tasks.

Segmentation and extraction of piece shapes in the layout diagram

The common format for layout diagram output is *dwg* or *dxf*, and the layout diagram needs to be converted into bitmap format for piece graphic processing. In this paper, 1 mm in the actual size corresponds to a pixel in the bitmap to intuitively reflect the size of the piece in the figure. In the layout diagram, each piece is a closed contour. Therefore, the extraction of piece graphics from the layout diagram is the process of extracting the closed contour, which is as follows:

- Read the layout diagram and convert it into a binary image: In a binary image, the pixel lattice of the image is all composed of 0 and 1, in which the black pixel value is 0 and the white pixel value is 1.
- Invert color: Invert the background color and piece color so that the pixel value corresponding to the background area is 0 and the pixel value corresponding to the piece contour is 1.
- Internal filling of the piece contour: After reversing the color, the background of the layout diagram is black, and the piece contour is white. Furthermore, fill all pieces with white so that all pieces have a pixel value of 1, including the piece contour and interior.
- Segmentation and storage of pieces graphics: after the processing in the previous step, each piece is a connected area composed of pixels with a value of 1, usually called the connected component of the binary image. After the treatment of the above work, the problem of extracting piece graphics is transformed into the boundary tracking of the connected component. The *bwboundaries* function in Matlab can be used to obtain the point coordinates of all contours of each piece. Furthermore, the piece graphics in png format is generated and saved.

After the complete segmentation and extraction of each piece in the layout diagram, the two piece can be matched directly in the similarity measurement algorithm. The segmentation and piece extraction of the layout diagram is shown in Fig. 6.

Identification of pieces sequence and angle

As one of the main techniques in the study of packing problems, the sequence and placement angle of pieces has been widely concerned. The combination of piece sequence and positioning can realize the optimal packing on the plate. In addition to the difference in the shape of pieces, there are differences in the size of pieces. The same scaling factor is used when transforming the piece graphics and layout diagram into bitmaps so that the size of pieces can be intuitively reflected in its area. The shape context algorithm^{45,60} can be used to mark the number and index of segmented pieces in the layout diagram to facilitate the recognition and processing of pieces sequence and rotation angle.

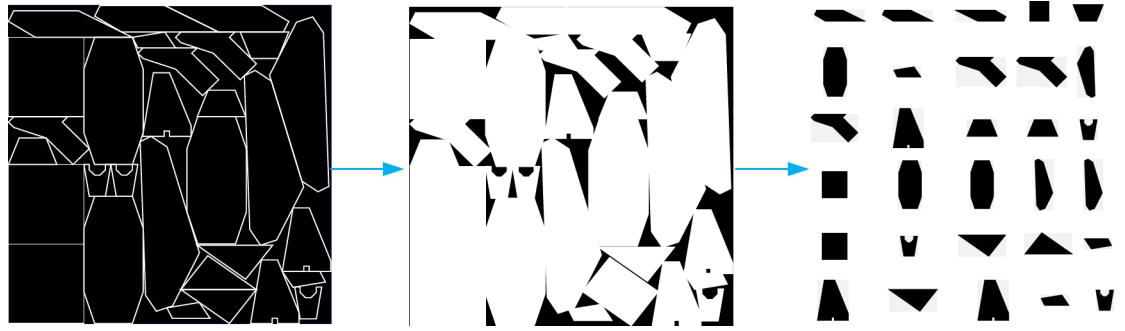


Fig. 6. The segmentation and piece extraction of the layout diagram.

- A. Pieces sequence identification of the layout diagram: Currently, the positioning algorithm for 2D irregular pieces packing includes the BL algorithm and the iBL algorithm, etc. The common feature of these algorithms is that pieces are placed in the bottom left position as far as possible during the packing. Therefore, the packing sequence of pieces can be inferred according to their positions in the layout diagram. The steps include: (1) Obtain the x-coordinate x_L of the leftmost point of all pieces in the layout diagram and the y-coordinate y_B of the bottommost point; (2) Select the piece at the bottom left of the layout diagram as the initially arranged piece, and update the contour of pieces already placed; (3) Search for the remaining pieces that are adjacent to the contour of the already arranged pieces, and select the piece with the smallest x_L as the next arranged piece. When x_L is equal, the piece with the smallest y_B is selected as the next arranged piece, and update the contour of already arranged pieces; (4) Repeat the third step until the packing sequence of all pieces is found and outputs the final packing sequence that to be arranged.
- B. Rotation angle identification of pieces: In the process of packing, the rotation angle of pieces is one of the factors that affect the packing utilization rate. Pieces are often set to rotate at fixed-angle steps. Common angle rotation steps are 30° , 45° , 60° , and 90° . From Fig. 6, the rotation angle is retained of piece graphics extracted from the layout diagram. Therefore, the rotation angle of pieces in the layout diagram can be obtained by comparing the piece graphics data extracted from the layout diagram with the original layout diagram. Assuming that the main axis direction angle extracted from the layout diagram is θ_1 , the original main axis direction angle of the piece is θ_0 . The rotation angle θ of the piece can be represented by Formula (13).

$$\theta = \theta_1 - \theta_0 \quad (13)$$

Reusability evaluation of packing tasks

The degree of repetition between the target task and the source task determines the proportion and the effect of transferable knowledge in some sense. Generally speaking, when the target task has a high similarity with the source task, the target task can obtain more knowledge and experience from the source task and has a high probability of inheriting the excellent packing effect from the source task. The reusability evaluation of packing tasks can intuitively express the amount of knowledge that the target task can inherit from the knowledge base of the source task.

Repetition degree of packing tasks

In two packing tasks, pieces with the same completion are called duplicate pieces, and pieces with similar shapes and sizes are called similar pieces. Assuming that the number of pieces in the packing task M is m , the number of pieces in the packing task N is n , and the number of duplicate pieces in tasks M and N is d , the repetition degree $G_R(M, N)$ of the packing tasks M and N can be obtained by Formula (14).

$$G_R(M, N) = \frac{2d}{m + n} \times 100\% \quad (14)$$

Similarity of packing pieces

The shape similarity of a piece is not equal to the similarity of a piece. The influence caused by the size gap should be considered when calculating the similarity of a piece. The calculation of the piece similarity proposed in this paper is based on the shape similarity of the twin neural network model, and the correction coefficient is used for accurate calculation. Set the similarity of the pieces in the layout diagram to E_V , set the shape similarity output by the twin neural network to E_C , and set the correction coefficient of piece size to ε . The calculation of the similarity of the pieces is shown in Formula (15). Define the areas of two pieces to be compared as s_1 and s_2 , respectively. Assuming $s_1 \leq s_2$, the size difference r of the piece can be expressed by Formula (16). When the areas of the two piece are equal, the size difference is 0. When the areas of two pieces differ greatly, the size difference r approaches 1. The relationship between the correction coefficient ε and the size difference degree r is shown in Formula (17).

$$E_V = \varepsilon \cdot E_C \quad (15)$$

$$r = 1 - s_1/s_2, \quad 0 \leq r \leq 1 \quad (16)$$

$$\varepsilon = (1 - r/(1 + r))^2 \quad (17)$$

Reusability evaluation of packing tasks

The reusability evaluation of the packing task can provide quantitative data when the target packing task inherits and reuses the packing knowledge from the source task, which lays a foundation for further knowledge transfer. The reusability evaluation of packing tasks is divided into the calculation of the similarity of pieces in the task and the calculation of the reusability between tasks. Assume that the number of pieces in task M is m , and the number of pieces in task N is n . Calculate each piece a_i ($i = 1, 2, \dots, m$) in task M and each piece b_j ($j = 1, 2, \dots, n$) in task N respectively to obtain the piece similarity matrix K_{ij} with size $m \times n$. The pseudo pieces matching strategy is introduced to solve the problem of the unequal number of pieces between different tasks. More specifically, pseudo pieces are used to supplement the number of pieces in the task with fewer pieces, and the similarity between the added pseudo pieces and any piece is 0. Assuming $m < n$, the piece matching relationship between tasks is shown in Fig. 7. After the pseudo pieces are added to task M , the size of the new piece similarity matrix K'_{ij} becomes $n \times n$. The Hungarian algorithm⁶¹ can be used to obtain the maximum sum R_{max} of the piece similarity between tasks, as shown in Formulas (18) and (19).

$$\max R = \sum_i \sum_j K'_{ij} \cdot y_{ij} \quad (18)$$

$$s.t. \begin{cases} \sum_i y_{ij} = 1, \quad j = 1, 2, \dots, n \\ \sum_j y_{ij} = 1, \quad i = 1, 2, \dots, n \\ y_{ij} = 0 \vee y_{ij} = 1 \end{cases} \quad (19)$$

When a_i matches b_j , y_{ij} takes the value 1. When a_i does not match b_j , y_{ij} takes 0. G_s represents the similarity of packing tasks, as shown in Formula (20). In certain cases, the degree of packing task repetition is equivalent to the degree of packing task similarity. Since duplicate pieces and similar pieces have different values for the reuse of packing knowledge, the weighted summation of the repetition degree and similarity degree of the source task and the target task can effectively evaluate the reusability of the historical task to the current packing task. Set the weight of the task repetition to γ , $\gamma \in (0, 1)$. Set the repetition degree to G_R and the reusability degree to G of packing tasks. The corresponding calculation is shown in Formula (21). To balance the proportion of duplicate pieces and similar pieces in the packing application, the value of γ in this paper is set to 0.3. When all similar pieces in task M and task N are duplicate pieces, the degree of similarity is equivalent to the degree of repetition of packing tasks. Formula (21) is used to evaluate the reusability of one or more similar source tasks, and as a basis for selecting the source task for knowledge transfer.

$$G_S = G_S(M, N) = 2 R_{max}/m + n \times 100\% \quad (20)$$

$$G = \gamma \cdot G_R + (1 - \gamma) \cdot G_S \quad (21)$$

For example, the number of pieces in the target packing task FJ200 is 142, as shown in Table 1. The existing source task packing diagram FJ400 and FJ600 are shown in Figs. 8 and 9, respectively. According to the proposed

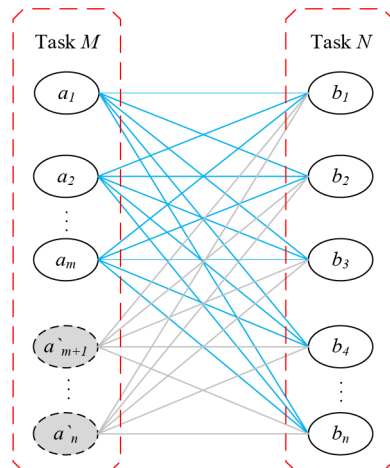


Fig. 7. Pieces matching between tasks.













Piece shape	Piece label	Number of pieces	Piece shape	Piece label	Number of pieces
	Shapes1_0	15		Shirts_2	8
	Shapes1_1	7		Shirts_3	15
	Shapes1_2	9		Shirts_4	15
	Shapes1_3	12		Shirts_5	15
	Shirts_0	8		Shirts_6	15
	Shirts_1	8		Shirts_7	15

Table 1. Pieces information in the packing task FJ200.

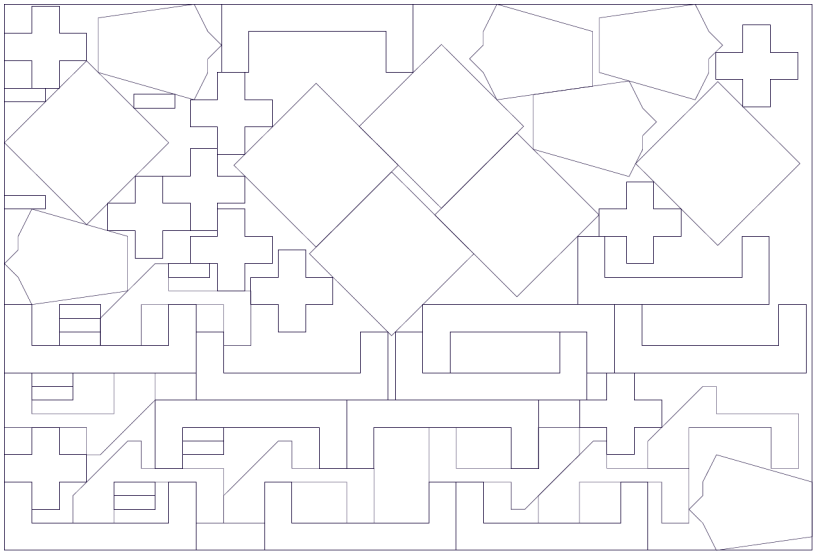


Fig. 8. Layout diagram FJ400 in the source task.

pieces extraction principle, pieces information in the packing diagram FJ400 and FJ600 can be obtained as shown in Tables 2 and 3, respectively. The results of similarity matching between the piece in the packing task FJ200 and pieces in FJ400 and FJ600 are shown in Table 4. From Tables 3 and 4, and 5, it can be seen that the number of pieces in the layout diagram FJ400 and FJ600 is 54 and 81, respectively, among which there are 54 and 57 duplicate pieces, respectively. Therefore, the repetition degree G_R (FJ200, FJ400) and similarity degree G_S (FJ200, FJ400) of the packing task FJ200 and the layout diagram FJ400 are 55.10%. The repetition degree G_R (FJ200, FJ600) and similarity degree G_S (FJ200, FJ600) of packing task FJ200 and layout diagram FJ600 are 51.12% and 57.38%, respectively. Furthermore, the reusability of layout diagram FJ400 and FJ600 in the source task to task target FJ200 is 55.10% and 55.50%, respectively. Therefore, pieces in the packing task FJ200 will preferentially inherit the sequence of corresponding matched pieces in FJ600, and the 2D irregular packing will be completed under the guidance of reinforcement learning.

Transfer operator and reinforcement learning search
Sequence transfer based on reusability evaluation of packing tasks

After the segmentation and extraction of the layout diagram of excellent cases, the corresponding piece data set can be obtained, which is also regarded as the source task of the piece packing. The source task composed of multiple pieces datasets are stored in the information storage module (ISM), which is the packing knowledge base. The target task is matched with the source task in the knowledge base, respectively, and the current source task is determined as the source knowledge of the piece similarity calculation. The transfer operator is used to transfer the piece sequence from the current source task to the target packing task, to improve the efficiency of

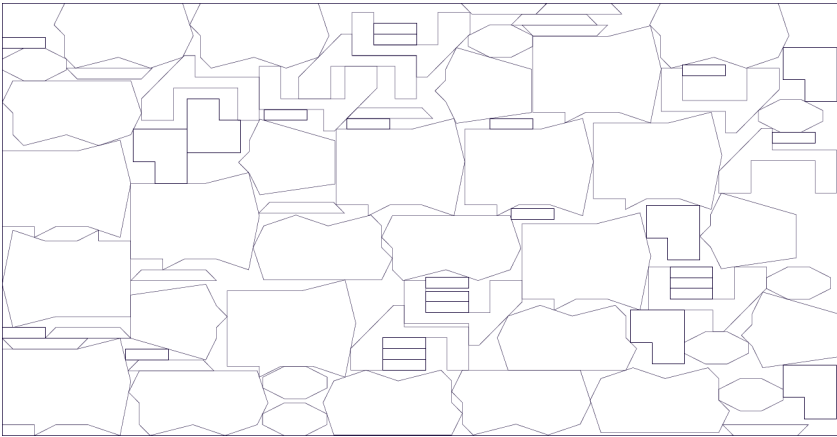


Fig. 9. Layout diagram FJ600 in the source task.







Piece shape	Piece label	Number of pieces	Piece shape	Piece label	Number of pieces
	Shapes1_0	12		Shapes1_3	10
	Shapes1_1	6		Shirts_1	6
	Shapes1_2	7		Shirts_6	13

Table 2. Pieces information in the layout diagram FJ400.









Piece shape	Piece label	Number of pieces	Piece shape	Piece label	Number of pieces
	Shapes1_2	9		Shirts_2	10
	Shirts_0	5		Shirts_4	12
	Shirts_1	11		Shirts_5	20
	Jakobs_0	6		Jakobs_1	8

Table 3. Pieces information in the layout diagram FJ600.

the packing search. The transfer operator in this paper refers to previous work⁴⁵. After the twin neural network calculates and matches the pieces in the current target task and the pieces in the source task, respectively, the transfer operator is used to inherit and transfer the corresponding packing sequence in the current source task. The successfully matched pieces between the target task and the current source task are stored in the set MC. Since the pieces in MC have certain packing sequences in the historical optimal layout case, the matching relationship between pieces in the target task and pieces in MC is used to optimize the packing sequence of the pieces in the target task. Assuming that the current source task set selected after the reusability evaluation of the packing task is $P = \{P_{f1}, P_{f2}, P_{f3}, P_{f4}, P_{f5}, P_{f6}\}$, and the target task set is $P_0 = \{P_{01}, P_{02}, P_{03}, P_{04}, P_{05}\}$, the matching result of the target task set P_0 is $MC = \{P_{f1}, P_{f2}, P_{f4}, P_{f5}, P_{f6}\}$. Pieces in the target task are matched with pieces in the MC one by one, and the packing sequence and rotation angle of the pieces in the MC are inherited. Finally, the packing sequence $\{P_{03}, P_{01}, P_{05}, P_{04}, P_{02}\}$ of the target task is obtained, as shown in Fig. 10.

Reinforcement learning search

The source task provides source knowledge for knowledge reuse and knowledge transfer of the target task. However, there are also certain cases where the pieces in the target task do not match the pieces in the source task at all or partially. For example, when MC is an empty set or when MC is not empty but has less number

Piece label in FJ200	Piece label in FJ400	Similarity of pieces	Number of pieces	Piece label in FJ200	Piece label in FJ600	Similarity of pieces	Number of pieces
Shapes1_0	Shapes1_0	100%	12	Shapes1_2	Shapes1_2	100%	9
Shapes1_1	Shapes1_1	100%	6	Shirts_0	Shirts_0	100%	5
Shapes1_2	Shapes1_2	100%	7	Shirts_1	Shirts_1	100%	8
Shapes1_3	Shapes1_3	100%	10	Shirts_2	Shirts_2	100%	8
Shirts_1	Shirts_1	100%	6	Shirts_4	Shirts_4	100%	12
Shirts_6	Shirts_6	100%	13	Shirts_5	Shirts_5	100%	15
--	--	--	--	--	Shirts_1	0	3
--	--	--	--	Shirts_0	Shirts_2	29.71%	2
--	--	--	--	Shirts_6	Shirts_5	93.62%	5
--	--	--	--	Shapes1_3	Jakobs_0	18.20%	6
--	--	--	--	Shirts_3	Jakobs_1	7.65%	8

Table 4. Similarity calculation of pieces.

Problem instance	Number of pieces types	Total number of pieces	Rotational constraints	Sheet width
200T-3041	17	38	90 incremental	2000
200T-3054	6	36	90 incremental	2000
M8-1007	28	91	90 incremental	2000
200T-3010	20	102	90 incremental	2000
FJ200	12	142	0, 180 absolute	40
Dighe1	16	16	0 absolute	100
Dighe2	10	10	0 absolute	100
Fu	12	12	90 incremental	38
Jakobs1	25	25	90 incremental	40
Jakobs2	25	25	90 incremental	70
Marques	8	24	90 incremental	104
Shapes0	4	43	0 absolute	40
Shapes1	4	43	0, 180 absolute	40
Shapes2	7	28	0, 180 absolute	15
Shirts	8	99	0, 180 absolute	40
Swim	10	48	0, 180 absolute	5752
Trousers	17	64	0, 180 absolute	79

Table 5. Instance information of the test data.

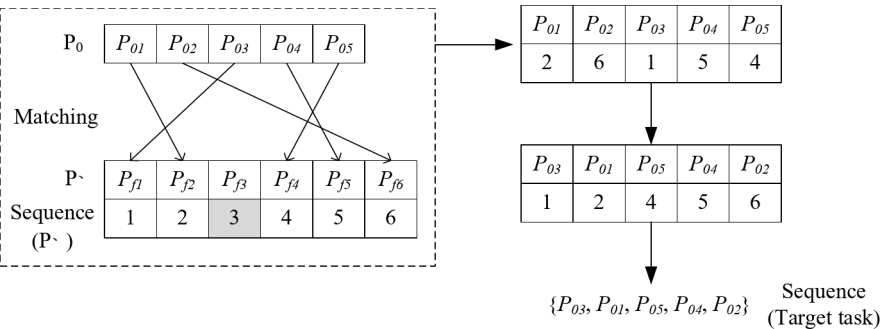


Fig. 10. Matching of the piece in the target task with the piece in MC.

of pieces than that in the target task. To improve the utilization rate of the packing, reinforcement learning with self-learning ability is designed to search the packing sequence. A hybrid positioning strategy based on fit and minimum envelope rectangle increment is combined to complete the packing of 2D irregular pieces. In this paper, the first visit Monte-Carlo reinforcement learning strategy^{6,62} is used to optimize the sequence of unmatched pieces in the target task. In the process of packing search based on reinforcement learning, a series of complete episodes are not only used to update status s and action a but also used to calculate reward r .

```

initialize, for all  $s \in n, a \in n, Q(s, a) \leftarrow 0$ 
for 1 to  $m$  do:
    choose  $a_i$  at  $s_{i-1}$  according piece generation strategy
    update  $s_i = a_i$ , generate an episode
    if  $i = n, H \leftarrow$  HPS algorithm, then  $r = C/H$ , else  $r = 0$ 
    for each pair  $(s, a)$ :
         $G \leftarrow$  first visit  $(s, a)$ 
        append  $G$  to  $Q(s, a)$ 

        
$$Q(s, a) \leftarrow Q(s, a) + \frac{1}{N(s, a)}(G - Q(s, a))$$

    end for
    update the optimal sequence
end for
output the optimal sequence

```

Algorithm 2. MCRL for sequence optimization.

The reward r is given after each episode is completed. Set $r = C/H$, where C is the constant, and H is the return packing height of each episode. For each state s in the episode, there is a discount return value G . Each time s is encountered, the average reward of the state can be calculated, represented by $Q(s, a)$. After each episode, the set of state and action of pieces will change with the change of the reward value, and these changes correspond to the change of packing sequence and utilization rate. Therefore, the current optimal sequence after each episode update represents a solution to the problem of 2D irregular pieces layout. The process of sequence optimization based on Monte-Carlo reinforcement learning (MCRL) is shown in Algorithm 2.

Among them, a_i and s_i are the action and state of i -th stage, respectively, n is the number of pieces optimized based on reinforcement learning, and m is the number of episodes. $N(s, a)$ represents the number of identical state-action pairs in multiple episodes. Overall, the KRIH algorithm integrates various techniques and methods, including segmentation of excellent layout diagram, extraction of packing knowledge, pieces similarity measurement, reusability evaluation of the packing task, reinforcement learning optimization, transfer mechanism design, and positioning algorithm improvement, etc. The flow of the KRIH algorithm is shown in Fig. 11.

Computational experiment and discussion analysis

Computational experiment

The project is written in Python, and tools such as Matlab are utilized. Computational tests of the proposed KRIH algorithm are performed on a computer with a 2.30 GHz AMD Ryzen 7 3750 H CPU with 4 kernels and 16 GB of RAM. The shape of 2D irregular pieces is mainly composed of polygons with concave and convex features. On the one hand, these polygons do not have flexible deformation and limb changes, nor do they need to be considered for the noise and interference in the image. On the other hand, the shape categories and quantity of the currently published standard dataset are insufficient for neural network training. Therefore, a 2D shape dataset mainly composed of polygons is constructed in the paper, in which 115 types are included, as shown in Fig. 12. The shape generator is used to generate 240 similar pieces graphics for each type of polygon according to different deformation degrees, and these 240 pieces graphics are used as a training set. That is, 27,600 pieces graphics are contained by the constructed data set. The binary cross entropy loss is taken as the loss function of the model. The batch size of the training is set to 32. When all piece graphics in the data set are traversed, it is recorded as an epoch. The number of iterations of the training data set is 786, and the number of epochs is set to 100. The Adam optimizer is used to optimize neural networks. The initial learning rate lr of the first 60 epochs is set to 0.001, and the decay index is 0.9. The initial learning rate lr of the last 40 epochs is set to 0.00001, and the decay index is 0.8. The loss curve of the training is shown in Fig. 13. The task repetition weight γ is set to 0.3. When all pieces in the target task are fully matched, the transfer operation can be carried out completely, and the condition for output the packing result is met. When pieces in the target task are partially matched, the MCRL algorithm is applied. The search episode is set to 300 times, The discount factor is set to 1, and the constant C is set to 100. The random exploration rate used to generate the piece sequence is 0.1. In the process of packing search based on reinforcement learning, the current minimum packing height returned is recorded and compared with the final packing height obtained. The minimum packing height is recorded as the final optimized packing height, and the corresponding optimization sequence is taken as the current optimal packing sequence. To test the performance of the proposed algorithm, Actual production instances from a Wuhan shipping company⁴⁵ and the benchmark test data from EURO Special Interest Group on Cutting and Packing (ESICUP, <https://www.esicup.org/>)

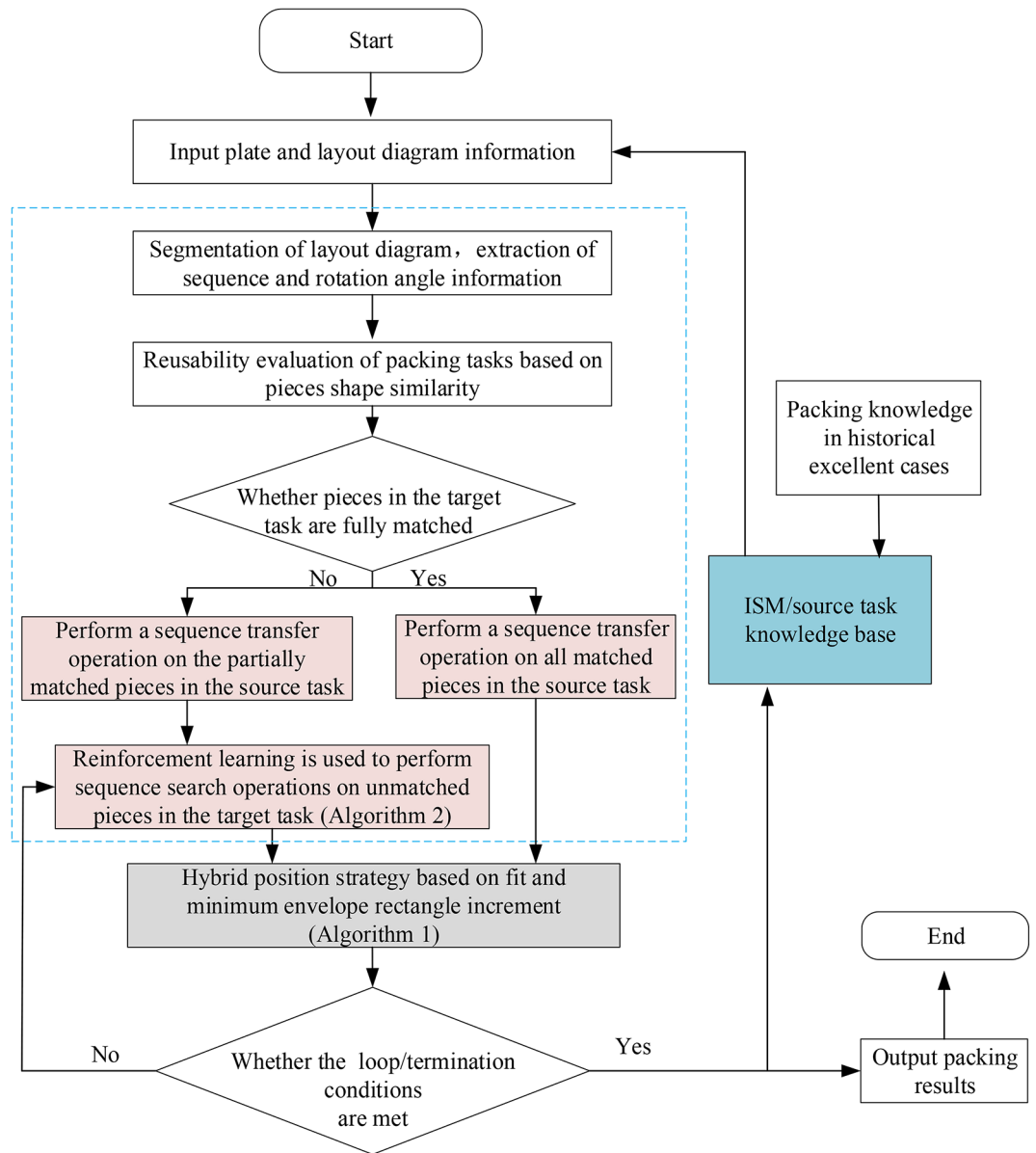


Fig. 11. The flow diagram of the KRIH algorithm.

www.euro-online.org/websites/esicup/data-sets/accessed on 10 March 2023) are used. Since most of the literature does not provide the layout diagram for all benchmark instances, the construction of the source task knowledge base is limited. Therefore, for the calculation on the benchmark instance, the historical excellent layout diagram is partly derived from the existing research achievements and partly generated by the method in the existing literature⁵⁹. Table 5 provides the instance information for the test data. The calculation results of relevant classical packing algorithms on different instances are shown in Table 6. The calculation results of the KRIH algorithm and the packing search algorithm (MC-HPS) fully utilizing Monte-Carlo reinforcement learning on different instances are shown in Table 7. For each instance, the MC-HPS algorithm and KRIH algorithm are run 10 times, and results are compared with those based on the relevant packing algorithm. Better average results based on the KRIH algorithm than those based on the MC-HPS algorithm are shown in bold. Better calculation results obtained based on algorithms MC-HPS and KRIH are represented by the underline compared with the results in the literature⁴⁵. The optimal result obtained by KRIH is highlighted in blue compared to the optimal result obtained by the relevant algorithm. With the instance as the horizontal coordinate and the utilization rate of the plate as the vertical coordinate, the point diagram of the optimal result obtained based on the KRIH algorithm and the optimal result obtained based on the classical packing algorithm is shown in Fig. 14.

By analyzing Tables 6 and 7; Fig. 14, it can be seen that the calculation results of different algorithms on different instances are divergent, while the calculation results on the same instance are similar, which indicates the complexity of instance calculation and the importance of algorithm design. The random HPS algorithm shows better computational results compared to the classical algorithm in some instances, demonstrating the

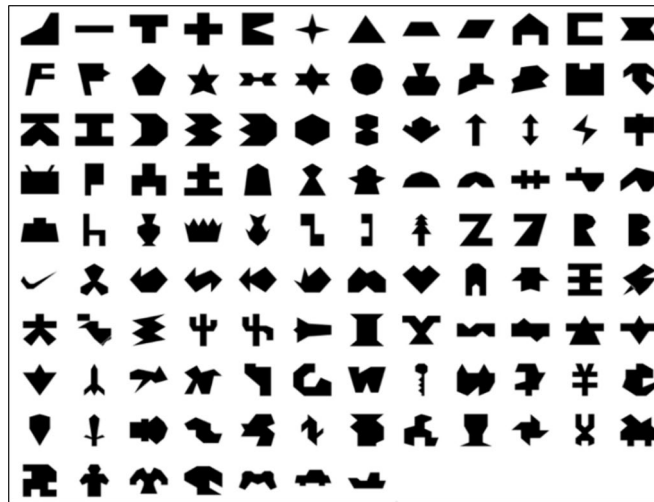


Fig. 12. Constructed 2D shape dataset.

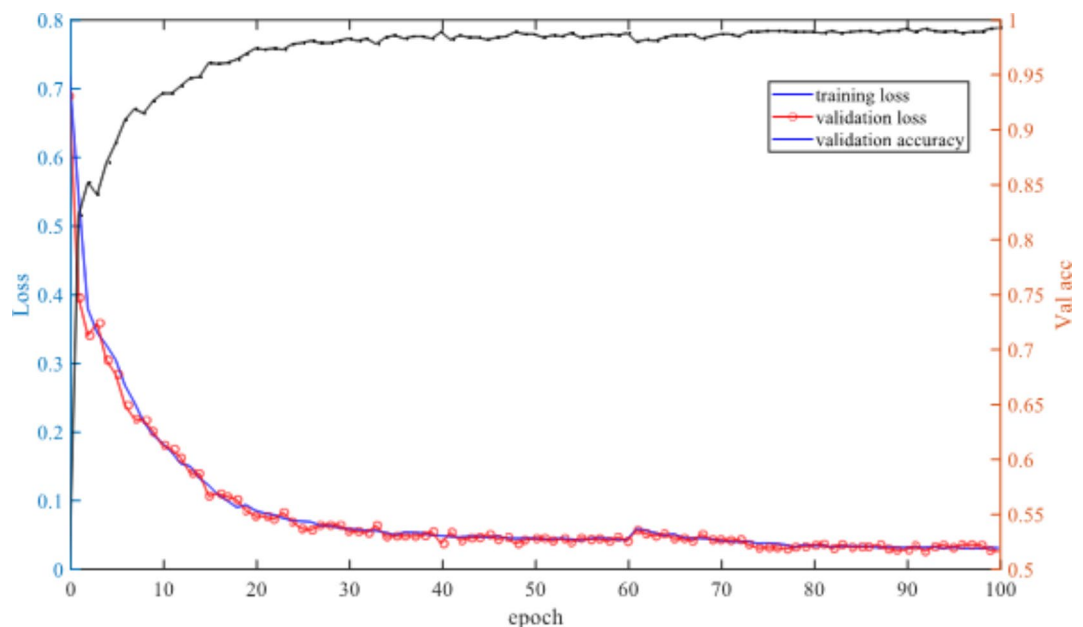


Fig. 13. The loss curve of the training.

applicability of the improved heuristic algorithm. The KRIH algorithm obtains 8 better or equal results on 16 instances, which shows good computational performance compared to the results based on listed classical algorithms. In comparison with results of the random search based on the HPS algorithm, the KRIH algorithm achieves better results on all instances except for instance 200T-3041. Compared with the MC-HPS algorithm, the KRIH algorithm obtained 12 better or equal average calculation results, indicating the effectiveness of the proposed algorithm in the reuse and transfer of source task knowledge. By comparing the MC-HPS algorithm with the OMCRL algorithm, it can be seen that the MC-HPS algorithm can get better results on all instances except for the Jakobs2, which shows the effectiveness of the improved heuristic algorithm. In addition, compared with the RKGA-NP algorithm, the KRIH algorithm can achieve a 100% utilization rate on instances Dighe1 and Dighe2, and has a shorter calculation time on all instances, which demonstrates the high efficiency of the proposed algorithm. In comparison with the GABL algorithm, the KRIH algorithm achieves better optimal results on 10 instances. Among them, the optimization effect is improved by 17.47% and 15.05% based on instance Shirts and instance Dighe2, respectively. Compared with the ST-PSO algorithm, although the KRIH algorithm only achieves better packing results on instance 200T-3010, the shorter calculation time is obtained on instances 200T-3041 and 200T-3054, which are 56.25s and 25.10s, respectively. For the instance FJ200 with 142 irregular pieces, the KRIH algorithm can reach a satisfactory solution category within an acceptable time. Furthermore, compared with classic algorithms, including the RKGA-NP algorithm and the GABL algorithm,

Problem instance	OMCRL ⁶		ST-PSO ⁴⁶			RKGA-NP ⁶³			GABL ⁶⁴	
	Best (%)	Avg (%)	Best (%)	Avg (%)	Time (s)	Best (%)	Avg (%)	Time (s)	Best (%)	Avg (%)
200T-3041	/	/	84.6	82.99	58.83	/	/	/	/	/
200T-3054	/	/	76.3	75.48	29.3	/	/	/	/	/
M8-1007	/	/	87.5	86.57	83.72	/	/	/	/	/
200T-3010	/	/	73.8	71.57	52.0	/	/	/	/	/
FJ200	/	/	/	/	/	/	/	/	/	/
Dighe1	77.02	76.94	73.6	72.44	5.44	100.00	99.96	2053	87.31	85.01
Dighe2	76.20	76.14	/	/	/	100.00	99.99	2291	86.92	81.42
Fu	77.50	77.23	/	/	/	/	/	/	83.82	82.22
Jakobs1	74.56	74.52	80.5	79.65	38.76	81.67	78.79	10,268	80.46	76.50
Jakobs2	68.20	68.00	/	/	/	79.75	77.90	12,205	71.24	69.69
Marques	77.72	77.61	84.4	83.67	3.90	84.47	84.31	6871	84.14	83.55
Shapes0	57.35	57.09	/	/	/	67.25	63.50	1826	57.11	56.59
Shapes1	57.35	57.15	65.9	65.24	3.61	66.37	65.98	3025	63.33	63.03
Shirts	78.71	78.56	/	/	/	/	/	/	71.04	70.65
Swim	64.41	64.23	/	/	/	73.59	70.78	14,352	67.61	66.81
Trousers	78.09	77.84	/	/	/	90.91	88.67	16,025	86.43	86.06

Table 6. Calculation results of relevant classical algorithms.

Problem instance	Random HPS	MC-HPS			KRIH		
	(%)	Best (%)	Avg (%)	Time (s)	Best (%)	Avg (%)	Time (s)
200T-3041	82.92	84.88	84.1	480.06	82.65	82.11	56.25
200T-3054	71.51	74.03	73.50	321.35	75.52	74.95	25.10
M8-1007	82.04	84.44	83.78	515.20	83.10	82.77	95.42
200T-3010	72.05	73.13	72.43	452.41	73.84	72.29	66.83
FJ200	76.25	77.89	77.07	6755.32	77.49	76.19	712.54
Dighe1	86.94	88.12	87.68	167.10	100.00	99.72	24.40
Dighe2	89.07	100	99.29	97.62	100.00	99.73	11.26
Fu	77.03	77.57	76.93	131.98	86.36	85.62	12.59
Jakobs1	79.50	81.66	80.49	528.55	81.67	79.09	106.72
Jakobs2	67.53	68.0	67.29	415.09	73.75	72.76	52.88
Marques	78.65	79.25	78.66	581.02	85.27	84.97	146.00
Shapes0	55.17	57.93	57.45	1800.49	61.38	61.19	205.28
Shapes1	63.33	65.52	65.03	3426.38	65.41	65.20	337.46
Shirts	78.71	79.64	79.07	5716.37	83.45	83.28	416.37
Swim	65.31	68.70	68.23	5748.21	70.77	70.78	471.89
Trousers	77.84	80.24	79.46	4380.50	85.81	85.76	363.33

Table 7. Experimental results of the KRIH algorithm. Significant values are in bold.

the KRIH algorithm shows that the difference between the optimal and average calculation results based on benchmark instances is almost no more than 2%, which indicates the stability of the proposed algorithm. The test experiment and statistical analysis show that the KRIH algorithm proposed in this paper can not only obtain better solutions on multiple instances but also has great advantages in solving efficiency, which has good application potential. Layout diagrams corresponding to the partial optimal results obtained by the KRIH algorithm are shown in Fig. 15.

Discussion and analysis

By analyzing the experimental results, it is not difficult to find that the KRIH algorithm can efficiently complete the pieces packing of different sizes and shapes due to the design of the algorithm. Compared with the existing transfer packing algorithm, the KRIH algorithm can not only break that only the same number of pieces can be transferred but also can transfer piece datasets, including the benchmark instance. In addition, the segmentation of the layout diagram, the matching of pieces, and the reusability evaluation of packing tasks provide support for the industrial application of the algorithm. From the loss curve of the training in Fig. 13, it can be seen that the convergence effect of the model is very good, which can provide an intuitive similarity evaluation of the pieces.

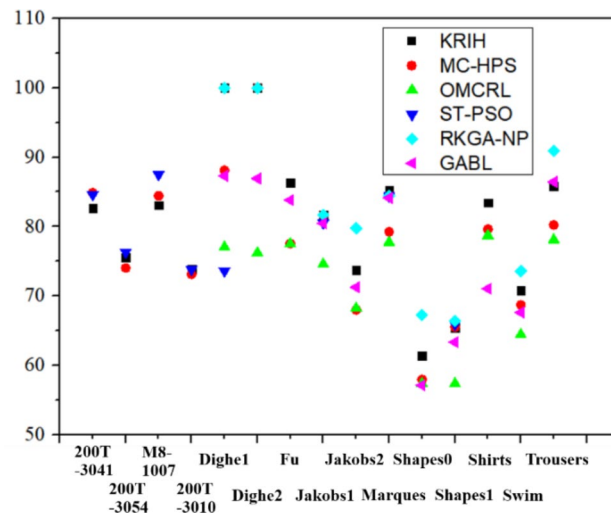


Fig. 14. The point diagram of optimal results obtained by each algorithm.

It can be seen from Table 7; Fig. 14 that only certain calculation results based on the MC-HPS algorithm are better than those based on the ST-PSO algorithm, and some results are worse than those based on random HPS. The possible reason is that the calculation based on the MC-HPS algorithm has no manual intervention, and the reinforcement learning search has a certain degree of randomness. In addition, the MC-HPS algorithm achieved better results than the OMCRL algorithm on 10 instances, which is related to the improved BL algorithm. The global situation and the special fit relationship between pieces are taken into consideration by the HPS algorithm, which can reduce wasted space. In the solving process based on the KRIH algorithm, most results reach or exceed the calculation result based on the MC-HPS algorithm, while some results (Jakobs2 and Shapes0, etc.) are not as good as the result based on the RKGA-NP algorithm. The possible reason is that the MC-HPS algorithm has not only a longer search time when the target task is completely different from the source task but also has a certain randomness. After a certain time of iterations, the MC-HPS algorithm may not be able to obtain the optimal packing scheme. Although the KRIH algorithm can transfer the packing sequences from excellent packing cases to improve the search efficiency. However, if only part of pieces in the layout diagram is identified and matched, some transferred pieces may not inherit the layout from the original packing diagram, thus resulting in an adverse impact on the packing utilization in the target task. In addition, the quality of the layout diagram in the source task, and the selection of the positioning algorithm for generating the layout diagram have a certain impact on the segmentation and sequence transfer of pieces. Therefore, the calculation result based on the KRIH algorithm may be not as good as those based on algorithms such as MC-HPS and RKGA-NP.

Although the KRIH algorithm can achieve a good packing effect in some instances, there are certain limitations. On the one hand, there is no large-scale training set for irregular pieces, and the self-generated training set has certain uncertainty in the recognition and matching of some piece shapes in benchmark instances, which may lead to unsatisfactory transfer effects. On the other hand, the transferred sequence is reversely derived according to the BL algorithm, and the transfer optimization of the sequence has limitations if the layout diagram used for transfer is not obtained through strict sequence optimization and position optimization.

Conclusions

In this paper, a new approach for the 2D plate packing problem using knowledge reuse and improved heuristic is proposed. The KRIH algorithm is guided by the manufacturing mode of batch production with variable variety in the heavy industry. It is an early attempt to solve 2D irregular packing problems using knowledge reuse and knowledge transfer. More specifically, the layout diagram based on historical excellent cases is segmented and recognized. The piece similarity measurement method based on twin neural networks is used to evaluate the similarity of pieces in the source task and target task. The reusability evaluation of packing tasks is designed to select appropriate source task knowledge in the ISM to guide the packing of target tasks. In addition, the transfer operator is developed to transfer the packing sequence from the source task so as to reuse the packing knowledge in the target task. To improve the packing efficiency, the BL algorithm is improved to complete the placement of 2D irregular pieces. For target task pieces that lack matching, Monte-Carlo (MC) based reinforcement learning algorithm is applied to supplement the search of the packing sequence. The computational experiments show that the proposed KRIH algorithm can not only complete the pieces packing of various sizes and shapes in an acceptable time but also obtain 8 better or equal results on 16 instances compared with results calculated by certain classical algorithms, shows good computational performance. In addition, the proposed KRIH algorithm can obtain a shorter calculation time on almost all instances, demonstrating a good solving efficiency. The experimental results not only demonstrate the effectiveness of the KRIH algorithm in optimizing complex packing problems, but also demonstrate its adaptability in handling diverse part shapes.

The experimental results not only provide new methodological support for the study of two-dimensional irregular piece packing problems, but also lay a theoretical foundation for practical applications in related

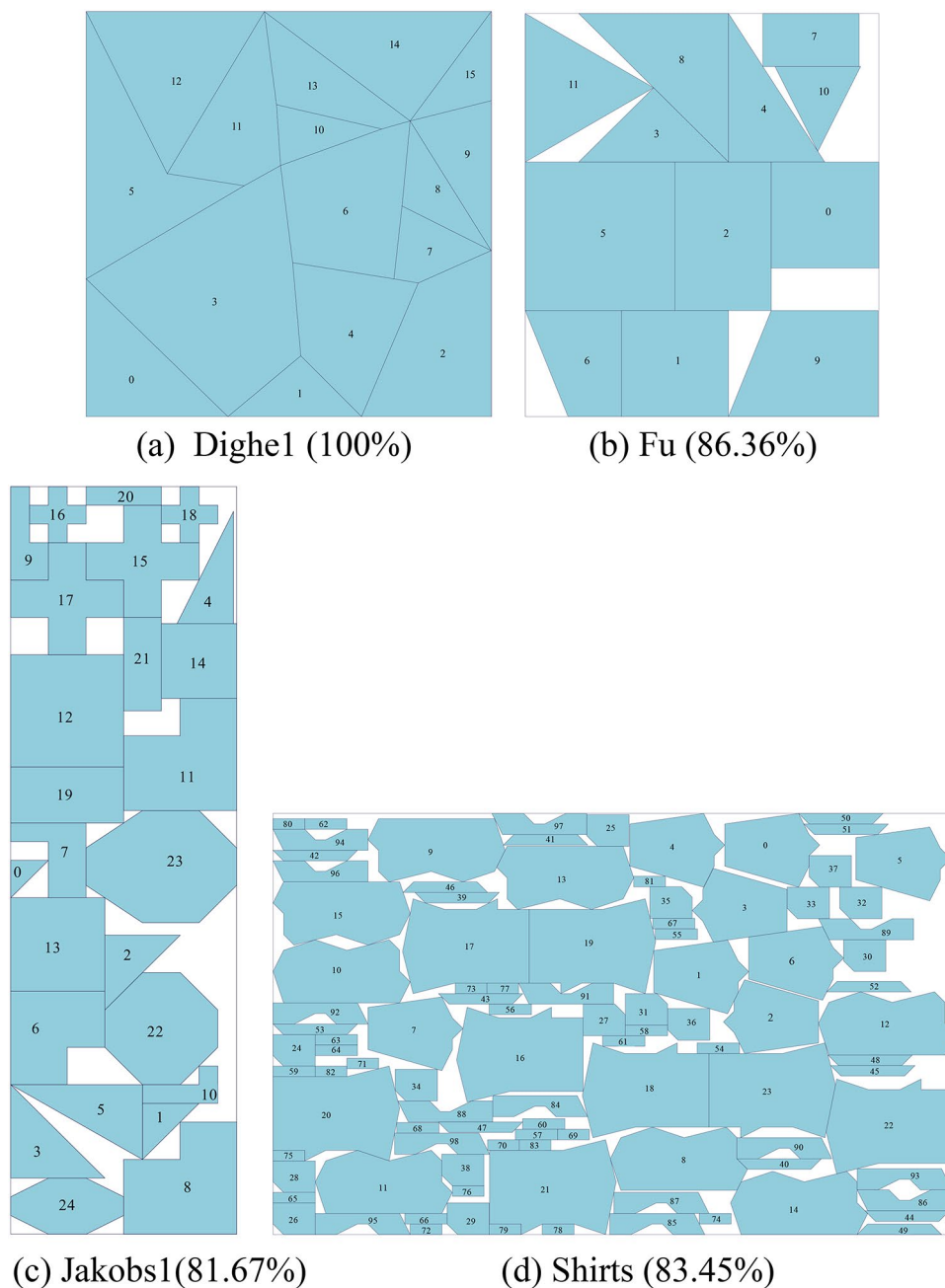


Fig. 15. Layout diagrams corresponding to the partial optimal results, (a) Dighe1 (100%) (b) Fu (86.36%), (c) Jakobs1 (81.67%) (d) Shirts (83.45%).

fields. The proposal of this algorithm is of great significance for reducing material waste, saving manufacturing costs, promoting environmental protection and improving economic benefits. It has a positive impact on the manufacturing optimization of complex heavy industrial equipment products and the material saving and consumption reduction of the whole society, demonstrating promising application prospects.

In future work, the training dataset based on the true pieces shape will be established, and the testing range of the proposed algorithm will be expanded. In addition, the deep network used to solve packing problems is also one of the directions worth studying. Furthermore, the advantages of the neural network will be fully utilized to explore more efficient transfer mechanisms to better solve 2D packing problems in the production.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 3 June 2024; Accepted: 28 November 2024

References

- Ke, Q. et al. Electric Vehicle Battery Disassembly sequence planning based on Frame-Subgroup structure combined with genetic Algorithm. *Front. Mech. Eng.*, **6**, 576642 (2020).
- Scher, W., Hau Ner, G. & Schumann, H. An improved typology of cutting and packing problems. *Eur. J. Oper. Res.* **183** (3), 1109–1130 (2007).
- Kucukyilmaza, T. & Kiziloz, H. E. Cooperative parallel grouping genetic algorithm for the one-dimensional bin packing problem. *Comput. Ind. Eng.*, **125**, 157–170. (2018).
- Fang, J. et al. Solving one-dimensional cutting stock problems with the deep reinforcement Learning. *Mathematics* **11** (4), 1028 (2023).
- Martello, S. & Monaci, M. Models and algorithms for packing rectangles into the smallest square. *Comput. Oper. Res.* **63**, 161–171 (2015).
- Fang, J. et al. A hybrid reinforcement learning algorithm for 2D irregular packing Problems. *Mathematics* **11** (2), 327 (2023).
- Duan, L. et al. A multi-task selected learning approach for solving 3D flexible bin packing problem. (2018).
- LI et al. One model packs thousands of items with recurrent conditional query learning. *Knowl. Based Syst.* **235**, 107683 (2022).
- Valle, A. et al. Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape. *Expert Syst. Appl.* **39** (16), 12589–12598 (2012).
- KOROUPI, F. & LOFTUS, M. *Accommodating Diverse Shapes within Hexagonal pavers*[J]291507–1519, (1991). 8.
- Yunlong, W. et al. Research on intelligent nesting algorithm for irregular ship parts based on no-fit-polygon. *Appl. Ocean Res.* **150**, 104108 (2024).
- BENNELL, A. B. E. Y. S. O. O. R. I. Y. A. R. P., Julia, A., MARTINEZ-SYKORA & Antonio Jostle heuristics for the 2D-irregular shapes bin packing problems with free rotation. *Int. J. Prod. Econ.* **195**, 12–26 (2018).
- Mundim, L. R., Andretta, M. & Queiroz, T. D. A biased random key genetic algorithm for open dimension nesting problems using no-fit raster. *Expert Syst. Appl.* **81** (SEP.), 358–371 (2017).
- Cherri, L. H., Cherri, A. C. & Soler, E. M. Mixed integer quadratically-constrained programming model to solve the irregular strip packing problem with continuous rotations. *J. Global Optim.* **72** (1), 1–19 (2018).
- Fischetti, M. & Luzzi, I. Mixed-integer programming models for nesting problems. *J. Heuristics*. **15** (3), 201–226 (2009).
- Guo, B. et al. Two-dimensional irregular packing problems: a Review. *Front. Mech. Eng.*, :79. (2022).
- Burke, E. K. et al. Irregular packing using the line and arc no-fit polygon. *Oper. Res.* **58** (4-part-1), 948–970 (2010).
- Chazelle The bottomn-left bin-packing heuristic: an efficient implementation. *IEEE Trans. Comput.* **100** (8), 697–707 (1983).
- Wang, X., Dai, C., Zhang, Q. & Tan, M. Layout Optimization of Square Parts Based on BL-IGA Algorithm[C]. In2023 38th Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 1051–1056.
- Mehta, P., Kumar, S. & Tejani, G. G. M. O. B. B. O. A Multiobjective Brown Bear optimization algorithm for solving constrained structural optimization Problems[J]. *J. Optim.* **2024** (1), 5546940 (2024).
- Nonut, A. et al. A small fixed-wing UAV system identification using metaheuristics. *Cogent Eng.* **9** (1), 2114196 (2022).
- Chibante, R. (ed) *Simulated Annealing: Theory with applications*[D] (BoD–Books on Demand, 2010).
- Prajapati, V. K., Jain, M. & Chouhan, L. Tabu search algorithm (TSA): A comprehensive survey[C]//2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE). IEEE, : 1–8. (2020).
- Jakobs, S. On genetic algorithms for packing of polygons. *Eur. J. Oper. Res.* **88** (1), 165–181 (1996).
- Li, M., SONG, C. F. & ZHOU, Z. K. Two dimensional irregular parts packing with particle swarm optimization. *J. South. Yangtze Univ.* **4** (3), 266–269 (2005).
- Kaya, E. et al. A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. *Eng. Appl. Artif. Intell.* **115**, 105311 (2022).
- Sato, A. K., Martins, T. C. & Guerra Tsuzuki, M. S. A study on GPU acceleration applied to 2D irregular packing problems. *Int. J. Comput. Integr. Manuf.*, (4/6):35. (2022).
- Abhishek, G. et al. Insights on transfer optimization: because experience is the best Teacher. *IEEE Trans. Emerg. Top. Comput. Intell.* **2** (1), 51–64 (2017).
- Iqbal, M., Browne, W. N. & Zhang, M. Reusing Building blocks of extracted knowledge to Solve Complex, large-scale boolean Problems. *IEEE Trans. Evol. Comput.* **18** (4), 465–480 (2014).
- Asta, S. et al. Generalizing hyper-heuristics via apprenticeship learning[C]. Evolutionary Computation in Combinatorial Optimization: 13th European Conference, EvoCOP, Vienna, Austria, April 3–5. (2013).
- Pelikan, M. & Hauschild, M. W. and Pier Luca Lanzi. Transfer learning, soft distance-based bias, and the hierarchical boa[C]. Parallel Problem Solving from Nature-PPSN XII: 12th International Conference, Taormina, Italy, September 1–5.
- Mazyavkina, N. et al. *Reinforcement Learning for Combinatorial Optimization: A survey*[J]134105400 (Computers & Operations Research, 2021).
- Xu, Q. et al. Multi-task optimization and multi-task evolutionary computation in the past five years: a brief review. *Mathematics* **9** (8), 864 (2021).
- Xue, X. et al. Evolutionary sequential transfer optimization for objective-heterogeneous problems. *IEEE Trans. Evol. Comput.* **26** (6), 1424–1438 (2021).
- Feng, L. et al. Memes as building blocks: a case study on evolutionary optimization + transfer learning for routing problems, *Memetic Comput.* **7**, 159–180 (2015).
- Gupta, A. et al. On the emerging notion of evolutionary multitasking: a computational analog of cognitive multitasking. *Recent. Adv. Evolutionary Multi-objective Optim.*, :139–157. (2017).
- DA et al. Evolutionary multitasking across single and multi-objective formulations for improved problem solving [C]. In: 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, 1695–1701. (2016).
- MANN, Y. O. G. A. T. A. M. A. D. & Gideon Efficient transfer learning method for automatic hyperparameter tuning[C]. In: Artificial intelligence and statistics. PMLR, pp. 1077–1085. (2014).
- O’NEILL et al. Common subtrees in related problems: a novel transfer learning approach for genetic programming[C]. In: 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE, 1287–1294. (2017).
- Yaddaden, A., Harispe, S. & Vasquez, M. Is transfer learning helpful for neural combinatorial optimization Applied to vehicle routing problems?. *Comput. Inform.* **41** (1), 172–190 (2022).
- Ning, F. et al. Manufacturing cost estimation based on similarity. *Int. J. Comput. Integr. Manuf.* **36** (8), 1238–1253 (2023).
- Patvardhan, C., Bansal, S. & Srivastav, A. Quantum-inspired evolutionary algorithm for difficult knapsack problems. *Memetic Comput.* **7**, 135–155 (2015).
- PILLAY, S. C. H. E. E. P. E. R. S. D. Nelishia. A study of transfer learning in a generation constructive hyper-heuristic for one dimensional bin packing[C]. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, pp. 1–7. (2021).
- ALODAT, M. Deep Transfer Learning and Intelligent Item Packing in Retail Management[C]. In: Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021. Singapore: Springer Nature Singapore, pp. 41–52. (2023).

45. Fang, J. et al. Sequence transfer-based particle swarm optimization algorithm for irregular packing problems. *IEEE Access*. **9**, 131223–131235 (2021).
46. Xu, X. F. et al. Transfer ants reinforcement learning algorithm and its application on rectangular packing problem. *Comput. Integr. Manuf. Syst. J.* **26** (12), 3236–3247. <https://doi.org/10.13196/j.cims.2020.12.006> (2020).
47. Zhang, F. et al. Packing optimization of rectangle workpieces oriented to variable-sized bin. *Comput. Integr. Manuf. Syst.* **21** (11), 2921–2928 (2015).
48. Xu, X. et al. Multi-objective robust optimisation model for MDVRPLS in refined oil distribution. *Int. J. Prod. Res.* **60** (22), 6772–6792 (2022).
49. Leao, A. A. et al. Irregular packing problems: a review of mathematical models. *Eur. J. Oper. Res.* **282** (3), 803–822 (2020).
50. Art, R. C. Jr *An Approach to the Two Dimensional Irregular Cutting Stock problem[D]* (Massachusetts Institute of Technology, 1966).
51. Burke, E. K. et al. Complete and robust no-fit polygon generation for the irregular stock cutting problem. *Eur. J. Oper. Res.* **179** (1), 27–49 (2007).
52. Liu Huyao. *Research of Two Dimensional Nesting Algorithm Based on no Fit Polygon [D]* (Shang Hai Jiao Tong University, 2007).
53. Keil, M. & Snoeyink, J. On the time bound for convex decomposition of simple polygons. *Int. J. Comput. Geom. Appl.* **12** (03), 181–192 (2002).
54. Bennell, J. A. & Song, X. A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums. *Comput. Oper. Res.* **35** (1), 267–281 (2008).
55. Li, Z. & Milenkovic, V. Compaction and separation algorithms for non-convex polygons and their applications. *Eur. J. Oper. Res.* **84** (3), 539–561 (1995).
56. Frat, H. & Alpaslan, N. An effective approach to the two-dimensional rectangular packing problem in the manufacturing industry. *Comput. Ind. Eng.* **148**, 106687 (2020).
57. Gu, J. et al. Recent advances in convolutional neural networks[J]. *Pattern Recogn.* **77**, 354–377 (2018).
58. Bromley, J. et al. *Signature Verification Using a Siamese time Delay Neural network[J]*6 (Advances in neural information processing systems, 1993).
59. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. (2014).
60. Belongie, S., Malik, J. & Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24** (4), 509–522 (2002).
61. Wright, M. B. Speeding up the Hungarian algorithm. *Comput. Oper. Res.* **17** (1), 95–96 (1990).
62. Jie, F. A. N. G. et al. A reinforcement learning algorithm for two-dimensional irregular packing problems[C]. In: Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence. pp. 1–6. (2021).
63. Du, B. Guo Xiaoqiang; Fang Jie; Wang Peng; Rao Yunqing. *A Hybrid Solving Algorithm on two-dimensional Irregular Graphics Nesting Problem*4739–45 (Forging & Stamping Technology, 2022). 3.
64. Pinheiro, P. R., Amaro Júnior, B. & Saraiva, R. D. A random-key genetic algorithm for solving the nesting problem. *Int. J. Comput. Integr. Manuf.* **29** (11), 1159–1165 (2016).

Author contributions

J.F.: conceptualization, methodology, validation, formal analysis, writing—original draft, writing—review and editing; X.C.: supervision, project administration. Y.R.: writing—review and editing; Y.P.: writing—review; K.Y.: writing—review; All authors have read and agreed to the published version of the manuscript.

Funding

This study was funded by the Natural Science Foundation of Wuhan Institute of Technology (grant number: K2024017), the Open Projects of Hubei Engineering Research Center for Intelligent Detection and Identification of Complex Parts (grant number: IDICP-KF-2024-12) and the National Natural Science Foundation of China (grant number: 52205536).

Declarations

Competing interests

The authors declare no competing interests.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Additional information

Correspondence and requests for materials should be addressed to X.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Scientific Reports is a copyright of Springer, 2024. All Rights Reserved.