

Received 17 July 2024, accepted 29 July 2024, date of publication 2 August 2024, date of current version 12 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3437729

RESEARCH ARTICLE

An Iterative Compression Method for the Two-Dimensional Irregular Packing Problem With Lead Lines

CHAO TANG¹, SHAOWEN YAO¹, LIMEI LU², SHIGANG ZHANG¹, AND LIJUN WEI¹

¹Key Laboratory of Computer Integrated Manufacturing System, School of Mechanical and Electrical Engineering, Guangdong University of Technology, Guangzhou, Guangdong 510006, China

²Shanghai Space Propulsion Technology Research Institute, Shanghai 201101, China

Corresponding author: Shaowen Yao (yaoshaowen97@gmail.com)

This work was supported in part by the Science Fund for Distinguished Young Scholars of Guangdong Province under Grant 2022B1515020076, and in part by the Natural Science Foundation of China under Grant 72271062.

ABSTRACT In industry, cutting various irregular pieces from a large raw material plate of a given size is often necessary to minimize the number of raw material sheets used. This problem is known as the two-dimensional irregular bin packing problem (2DIBPP). An iterative compression algorithm is proposed to address the irregular packing problem in the sheet metal industry, considering lead lines to maximize raw material sheet utilization. Firstly, three methods of lead lines pre-processing are proposed to effectively transform lead lines constraints into non-overlapping constraints between pieces. Secondly, an improved greedy heuristic, incorporating the sticking-edge and insertion-space strategies, is designed to obtain an initial solution for compact packing. Finally, through the iterative compression strategy, the occupied space of the pieces is continuously contracted to further enhance raw material sheet utilization. The efficiency of the proposed algorithms is demonstrated through testing and analysis of real-world instances from industry. The lead lines processing strategy and algorithm presented in this paper effectively resolve the irregular packing problem associated with lead lines, demonstrating their utility in industrial production.

INDEX TERMS Sheet metal industry, irregular packing, lead lines, iterative compression algorithm.

I. INTRODUCTION

The irregular packing problem [1] is widespread in industries such as apparel manufacturing and sheet metal cutting, where raw material sheets need to be cut into different pieces to maximize material utilization. In large equipment manufacturing companies, it is often necessary to cut a given lot of irregularly shaped pieces from a defined size of raw material sheet with the goal of minimizing the number of raw material sheets used. The problem is known as the two-dimensional irregular bin packing problem (2DIBPP). However, in the actual production process, not only is it crucial to efficiently utilize raw material sheets, but also the constraints of the cutting process must be taken into full consideration. Among these, the issues of inconvenient

packing and reduced sheet utilization arising from the inclusion of lead lines are particularly prominent. In the production process, large cutting machines usually use flame cutting and plasma cutting technology. To prevent piece defects resulting from the melting of the material at the tool cutting starting point due to high temperature, the processor will add lead lines to each piece to ensure that the position of the tool cutting starting point does not directly touch the edge of the piece. Therefore, the constraints imposed by the lead lines must be considered during the nesting process. Fig. 1 provides a detailed example of lead-in and lead-out lines. Fig. 1(a) illustrates a single piece featuring these lines, while Fig. 1(b) shows the result of piece packing. In this process, the cutting tool enters at the tail of the lead-in line and exits at the arrow of the lead-out line, adhering to the constraints imposed by the lead lines.

The associate editor coordinating the review of this manuscript and approving it for publication was Haidong Shao¹.

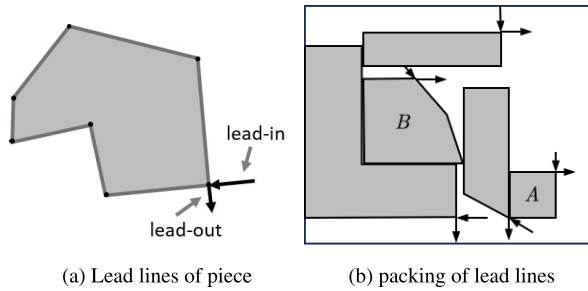


FIGURE 1. Lead lines cases.

When the lead-in and lead-out lines are integrated into the piece contour, the problem aligns with the 2DIBPP. The irregular packing problem that includes lead lines is a variant of 2DIBPP, as most 2DIBPP methods are applicable to solving the 2DIBPP with lead lines. In recent years, research on 2DIBPP has predominantly focused on the finite-rotation case [2]. López-Camacho et al. [3] employed the Djang and Finch heuristic to achieve better results for convex polygon instances than previously possible. Liu et al. [4] introduced a heuristic that combines a first-fit diminishing strategy for assigning irregular pieces to sheets with a swapping-piece method to enhance solutions for finite rotation 2DIBPP. Based on this, Zhang et al. [5] proposed a waste minimization strategy for piece assignment, alongside a hot-start and iterative doubling strategy to expedite the search process, resulting in significant improvements on classical cases. Additionally, numerous other methods have been utilized to address this problem, including scanline-based algorithms [6], genetic algorithms [7], hybrid heuristics [8], and hyper-heuristics [9].

In the realm of packing problems constrained by specific processes, researchers have dedicated their efforts to exploring the irregular packing problem associated with rectangular raw material sheets of varying sizes. Abeysooriya et al. [10] have introduced efficient local search heuristics to tackle the packing problem, achieving notable advancements. Additionally, they devised a heuristic algorithm based on algorithm of Jostle specifically tailored for the arbitrary angle problem. This algorithm includes a mechanism to identify promising subsets of angles based on the current arrangement of pieces within partial solutions. Yao et al. [11] proposed an iteratively doubling binary search strategy aimed at optimizing the utilization of differently shaped sheets over successive iterations. Their approach has demonstrated significant efficacy in real-world production. In addition to this, Yao et al. [12] conducted study on the 2D non-guillotine cutting problem with defect constraints. They applied decomposition of combinatorial benders to address the problem and achieved good results. Han et al. [13] and Martinez-Sykora et al. [14] extended their research to incorporate free rotation and guillotine constraints, addressing specific applications within the glass industry with innovative methodologies. Furthermore, Bennell et al. [15] raise a beam search heuristic to solve the packing problem. In this approach, each node represents a partial

solution, focusing on efficient arrangement strategies for raw sheets.

Given the current state of research, a significant portion of the literature focuses on the 2D irregular packing problem without incorporating process constraints. Among the studies that do consider process constraints, the examination of free rotation angles and guillotine constraints is more prevalent. However, there is a relative scarcity of research addressing the introduction of cutting process constraints, particularly those related to lead-in and lead-out lines. The presence of these process constraints poses a unique challenge, as existing methods for irregular packing cannot be readily applied to scenarios involving lead-in and lead-out lines. Moreover, the inclusion of these lines directly impacts the quality of the produced pieces. If these constraints are overlooked or mishandled, it can result in defective pieces or pieces that are aligned but not feasible for cutting. To address these challenges and ensure high-quality pieces while optimizing raw material utilization, it is imperative to develop an efficient algorithm for row sample placement that accounts for these critical process constraints.

The rest of this paper is organized as follows. Section II provides a formal description of the problem and details the calculation of utilization. Section III describes the geometric tools used in this paper. Section IV presents the piece lead lines handling methods. Section V explains the method and process of overlap minimization from pieces. Section VI introduces the proposed generating initial solution and iterative compression algorithm. Section VII details the experiments conducted, and Section VIII concludes the paper.

II. PROBLEM DESCRIPTION

The irregular packing problem in two dimensions considering lead lines are defined as follows: given a set of irregular pieces to be cut $P = \{p_1, \dots, p_n\}$ and a raw material sheet of length L and width W . The irregular packing problem is defined as follows. For piece p_i consists of a contour C_i , a lead-in line I_i , and a lead-out line O_i , where the contour, the lead-in line, and the lead-out line are represented using a point set. This is shown in Fig.1(a), where the arrow of the lead-in line points to the piece and the arrow of the lead-out line points to the piece. The piece is allowed to rotate by four angles during the packing process, where the set of angles at which the piece rotates is $A = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. It is necessary to pack all the pieces in the set P into the raw material sheets to maximize the utilization of the raw material. The following conditions need to be met during the packing process: (1) The pieces must not exceed the layout



FIGURE 2. An example of sheet layout.

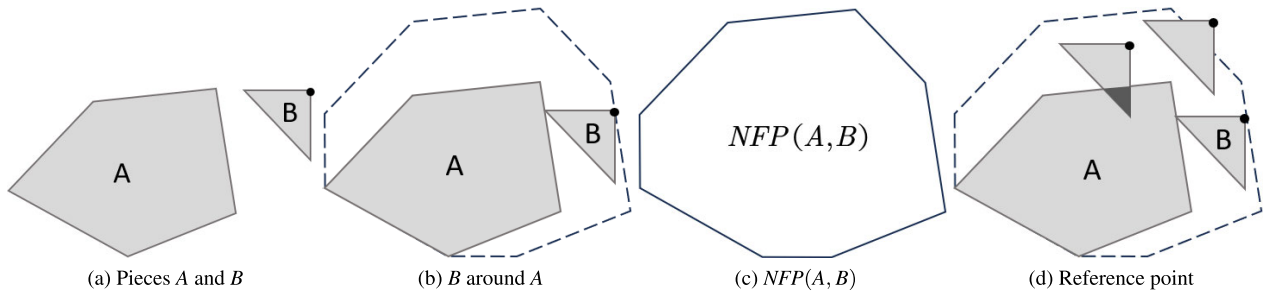


FIGURE 3. No-fit-polygon(NFP) for two irregular pieces A and B.

area of the raw material sheet; (2) No two pieces may overlap; (3) A certain distance must be maintained between pieces to ensure their integrity during cutting; (4) The edges of the pieces must maintain a certain distance from the boundaries of the raw material sheet to avoid defects; (5) The lead-in and lead-out lines of the pieces must not intersect with any other pieces to prevent compromising the integrity of other pieces during the cutting process.

During the packing process, the last sheet is often not completely filled. Calculating the average utilization solely based on the number of sheets can obscure the actual differences between algorithms. To more accurately measure the effectiveness of the algorithms, this paper calculates the utilization of the sheet as follows. In Fig.2, the used area of the last sheet is $S_{ulast} = S - S_e$, where S is the total area of the last sheet, S_e is the area of the rightmost endpoint of all the pieces further to the right, and the area of the pieces in the last sheet is $S_{plast} = S_{pa} + S_{pb} + S_{pc}$, which defines that the area of the first $n-1$ sheets is $(n-1)S$, and that of the pieces on the first $n-1$ sheets is S_p . $R_u = (S_{plast} + S_p) / ((n-1)S + S_{ulast})$, where n is the number of raw sheets utilized. This utilization prevails not only considers the complete use of sheets, but also fully considers the utilization of the last sheet in rows. Thus, the performance of the algorithm can be fully reflected.

III. GEOMETRIC TOOLS

A. NO-FIT-POLYGON

No-fit-polygon (NFP) serves as a fundamental geometric tool in addressing overlap detection between polygons within the irregular packing problem. Specifically, if the reference point of one polygon lies inside the critical polygon of another polygon, it indicates overlap between the two polygons; otherwise, no overlap exists.

Taking Fig.3 as an example, the process of generating $NFP(A, B)$ between two polygons A and B can be described as follows. Polygon A is considered a fixed polygon, while polygon B is treated as a tracking polygon allowed to slide along the outer contour of polygon A. Throughout this sliding process, polygon B maintains contact with the outer contour of polygon A, and the trajectory of reference point of polygon B is recorded. These trajectories of reference point of polygon B are then compiled to form polygon C, denoted as $NFP(A, B)$. For a detailed description of the NFP generation

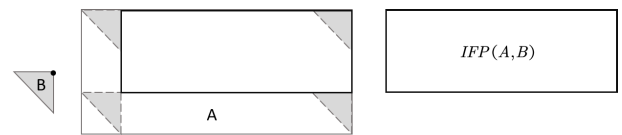


FIGURE 4. Inner-fit-polygon(IFP) for two irregular piece A and bins.

process, refer to the work of Bennell et al. As depicted in Fig.3, if the reference point of polygon B lies within $NFP(A, B)$, polygons A and B overlap; otherwise, they do not overlap.

To ensure the piece remains within the sheet during the overlap removal process, an Inner Fit Polygon (IFP) is defined for both the piece and the sheet. The IFP helps determine if the piece stays within the packing area of the sheet. If the reference point of the piece is not inside or on the IFP, it indicates the piece extends beyond packing area of the sheet. Conversely, if the reference point is inside or on the IFP, the piece remains within the packing area of the sheet. The process of generating the IFP is depicted in Fig.4. Initially, a reference point is selected from the piece. The piece is then moved along the inner edge of the sheet in a circular path. The region traced by the movement of the reference point of the piece forms the IFP. This polygon serves as a boundary to ensure the piece stays within the sheet during operations such as overlap removal.

B. OVERLAP DEPTH

In this paper, achieving a viable layout plan necessitates that pieces neither overlap nor protrude beyond the boundaries of the sheet. The concept of overlap depth is employed to assess the feasibility of the packing scheme. Overlap depth refers to the minimum distance required to separate two intersecting pieces or to fully accommodate a piece within the sheet when it extends beyond the interior of the sheet. For further elaboration on overlap depth, interested readers are referred to the works of Agarwal et al. [16], Abramson et al. [17] and Kim et al. [18]. Here are the definitions for clarity:

Definition 1: A polygon is represented by a set of points p . Given a polygon p and a translation vector $v = (v_x, v_y)$, denote the translation function by \oplus , so the translation process of p can be defined as: $p \oplus v = \{(ps_x + v_x, ps_y + v_y) | ps \in p\}$.

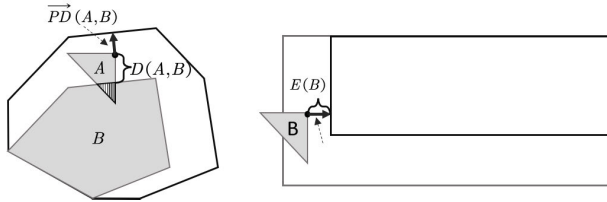


FIGURE 5. Overlap vectors and overlap values.

Definition 2: Given a polygon p and a rotation angle r , with the coordinate origin as the center of rotation, the rotation function of the polygon can be defined as: $p(r) = \{(ps_x \times \cos(r) + ps_y \times \sin(r), -ps_x \times \sin(r) + ps_y \times \cos(r)) | ps \in p\}$.

For a given piece a and piece b , the overlap distance between them can be denoted by $P(p_a, p_b) = \min\{\|v\| \mid p_a \cap (p_b \oplus v) = \emptyset, v \in \mathbb{R}^2\}$. If $P(p_a, p_b) = 0$, it means that pieces p_a and p_b are non-overlapping. Similarly, the overlap distance between piece b and sheet p can be expressed using $\bar{p}(b, p) = \min\{\|v\| \mid (p \oplus v) \in b, v \in \mathbb{R}^2\}$, which indicates that piece b is completely placed inside sheet p if $\bar{p}(b, p) = 0$. The overlap depth can be calculated from the overlap distance. The overlap vectors and overlap depths of the pieces are represented in Fig.5. This represents the relationship between the pieces, $\vec{PD}(A, B)$ represents the piece overlap vectors, and $D(A, B)$ represents the overlap depths; and Fig.5 represents the relationship between the pieces and the sheet, $\vec{PE}(B)$ represents the overlap vectors, and $E(A, B)$ represents the overlap depth.

After the overlap depth is obtained, the overlap value between two pieces can be further calculated, and in this paper, the overlap value is expressed by using the Euclidean Vander number de square of the overlap depth. The overlap value of two pieces p_k and p_l is expressed as (1), where R is a rotation angle vector storing the displacements of all the pieces; $v_i = (x_i, y_i)$ denotes the translation of piece p_i from the original coordinates of the reference point to the right by x_i and to the upward by y_i ; $p_i(r_i)$ denotes the rotation of piece p_i by r_i degrees, and the new coordinates of the points after the rotation of each point of piece p_i by r_i ; and $|C_i|$ is the set of contour points of p_i the number of points of p_i .

$$f_{kl}(R, V) = \|\vec{PD}(p_k(r_k) \oplus v_k, p_l(r_l) \oplus v_l)\|^2, \quad 1 \leq k < l \leq n \quad (1)$$

$$g_{kl}(R, V) = \|\vec{PE}(p_k(r_k) \oplus v_k, M(W, L))\|^2, \quad 1 \leq k \leq n \quad (2)$$

Similarly, the overlap of piece p_i with the raw material is expressed as (2), with M denoting the principle sheet, W denoting the width of the sampling area for the raw material sheet, and L denoting the length of the sampling area.

IV. PIECE LEAD LINES HANDLING METHODS

In this paper, three methods are proposed to deal with lead lines, namely, the shape expansion method, the local deformation method and the lead removal method of

lead lines. The first two methods optimize by transforming the piece shape from one with lead lines to a standard irregular polygon. The third method involves removing overlaps between lead lines and the piece through translation.

A. SHAPE EXPANSION METHOD

The shape expansion method combines the lead line points with the contour of piece to enlarge its shape in Fig.6(a). This transforms polygon A with lead lines into a simpler polygon A' , similar to solve a standard irregular packing problem. Without flaring, piece B can be placed closer to piece A in Fig.6. However, using the flaring strategy (creating A'), piece B must be positioned farther away, leading to increased material wastage.

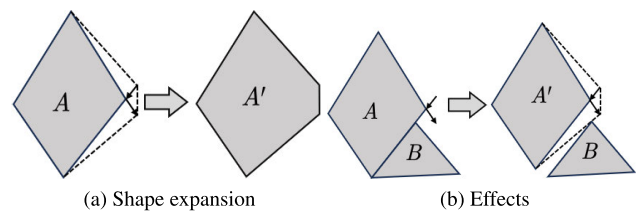


FIGURE 6. Shape expansion and effects.

B. LOCAL DEFORMATION METHOD

The piece local deformation method simplifies handling lead lines by covering them with two envelope rectangles that merge with the rest of the piece. These rectangles are considered part of the piece during packing. If piece B intersects the envelope rectangles of lead lines in piece A , the packing scheme is deemed invalid. In Fig.7(a), the method uses the two smallest envelope rectangles to cover the lead lines of piece A , transforming it into A' . This process can be optimized to reduce the area of the envelope rectangles. Firstly, the rectangles are aligned parallel to the lead lines in Fig.7(b). Then, a small extension is added to the adjoining portion of the piece to ensure the outline of the envelope rectangles completely overlaps the piece. Finally, the two piece contours are merged to form a new polygon for packing. This method is less wasteful than piece flaring, as using the merged piece for packing minimizes the impact of lead lines on packing utilization.

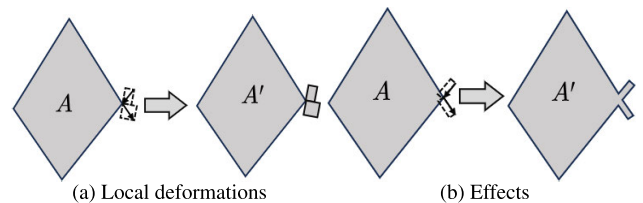


FIGURE 7. Local deformations and effects.

C. LEAD REMOVAL METHOD

The lead removal method directly substitutes lead lines constraints into the piece overlap calculation and consists of

three steps. The steps are as follows. Firstly, during piece processing, add the set of lead-in line points to the outline of the piece properties. Then, before placing a piece into the packing area, check if the lead-in lines of the current piece intersect with any already placed pieces. Finally, if an intersection is found, perform an overlap removal operation. Piece *B* overlaps the lead lines of piece *A* in Fig.8. Before eliminating the overlap, it is necessary to determine the shortest vectors that move the lead-in and lead-out lines out of the piece, respectively. The shortest vectors \vec{PX} , \vec{PY} and \vec{PZ} indicate the three directions in which the lead-in line moves out of piece *B*, with the shortest length of \vec{PX} used as the overlap vector of the lead-in line and piece *B*. And the shortest vectors \vec{QE} , \vec{QF} and \vec{QG} indicate the three directions in which the lead-out line moves out of piece *B*, with the shortest length of \vec{QE} used as the overlap vector of the lead-out line and piece *B*. The lead-in line and lead-out line are finally moved out of piece *A* separately. The overlap vectors of the lead-in line and the lead-out line with piece *B* are vectorially summed to obtain the total overlap depth \vec{PQ} . If a line segment in the lead lines does not intersect with other pieces, the overlap depth corresponding to that vector is (0,0). Finally, (3) is used to calculate the overlap value of the lead-in lines for p_k and p_l . This overlap value accounts for not only the overlap value of the lead lines of piece p_k with piece p_l , but also the overlap value of the lead lines of piece p_k with piece p_l .

$$h_{kl}(R, V) = \|\vec{PQ}(p_k(r_k) \oplus \mathbf{v}_k, p_l(r_l) \oplus \mathbf{v}_l)\|^2 + \|\vec{PQ}(p_l(r_l) \oplus \mathbf{v}_l, p_k(r_k) \oplus \mathbf{v}_k)\|^2, \quad 1 \leq k < l \leq n \quad (3)$$

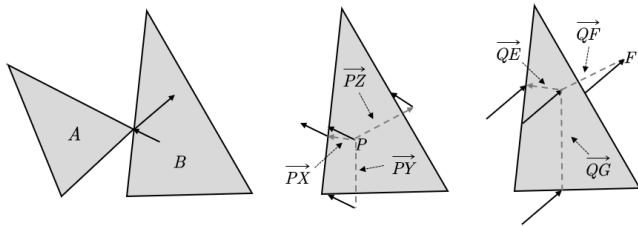


FIGURE 8. Lead removal method.

Fig.9 illustrates a simplified process for direct carry lead lines packing. Knowing the positions of piece *A* and piece *B* in the current legal solution, when placing piece *C*, piece *C* will first find a legal position without considering the lead lines using the conventional packing method. At this time, piece *C* overlaps with the lead lines of piece *A*, and the lead lines of piece *C* intersect with piece *B* in Fig.9(a). To resolve these overlaps, the vector \vec{PQ} can be computed based on the parameters, and piece *C* is translated according to this vector. After the translation, the legality of the new position of piece *C* is verified. The result shown in Fig.9(b). In cases where a feasible solution cannot be found, as illustrated in Fig.9(c), piece *D* cannot find a suitable location near its current position. When this occurs, the piece is removed, and

the upper right corner is re-traversed to find another feasible placement area.

V. OVERLAP MINIMIZATION

All feasible solutions to the packing problem require that there be no overlap between pieces and that the pieces are located within the raw material sheet. If any overlap exists between pieces, the packing solution is not feasible. Therefore, pieces must be moved to eliminate overlaps and achieve a feasible solution. In this paper, the total overlap value is defined in (4), where $f_{kl}(R, V)$, $g_k(R, V)$ and $h_{kl}(R, V)$ are derived from (1), (2), and (3), respectively, representing the overlap values of all pieces combined. It is important to note that (3) does not need to be considered if the lead lines of the pieces are managed using a method other than that described in Section IV-C.

When $Overlap(R, V) = 0$ for the packing result, the obtained solution is feasible. The process of removing overlap does not consider angular rotation of the pieces but only reduces the overlap value by moving the pieces. Therefore, the only variable in the formula is the translation vector of the *V*. Consequently, the process of removing overlap can be converted into an unconstrained nonlinear programming model to solve for the solution in (5).

$$Overlap(R, V) = \sum_{1 \leq k < l \leq n} f_{kl}(R, V) + \sum_{1 \leq k < l \leq n} g_{kl}(R, V) + \sum_{1 \leq k < l \leq n} h_{kl}(R, V) \quad (4)$$

$$Minimize O(V) = \sum_{1 \leq k < l \leq n} f_{kl}(V) + \sum_{1 \leq k < l \leq n} g_{kl}(V) + \sum_{1 \leq k < l \leq n} h_{kl}(V) \quad (5)$$

If the final result of $O(V) = 0$, it indicates that all overlaps have been successfully removed, resulting in a feasible layout solution. However, if the overlap value is not 0, it implies that the current layout cannot be adjusted to achieve a feasible solution solely by moving the pieces. In such cases, it becomes necessary to either remove some pieces from the current layout or modify the rotational state of the pieces to resolve the overlaps and achieve feasibility. Algorithm 1 provides pseudo-code for overlap minimization.

VI. ALGORITHM

In this paper, an algorithm is devised for addressing the irregular packing problem with lead lines, which consists of two main components. Firstly, an enhanced greedy search algorithm is introduced to discover the initial solution for the packing scheme. Subsequently, the initial solution generated is refined using an improved piece exchange algorithm to achieve further optimization of the packing solution.

A. GENERATION THE INITIAL SOLUTUIN

One commonly used heuristic strategy in the initial solution generation phase of irregular packing is the

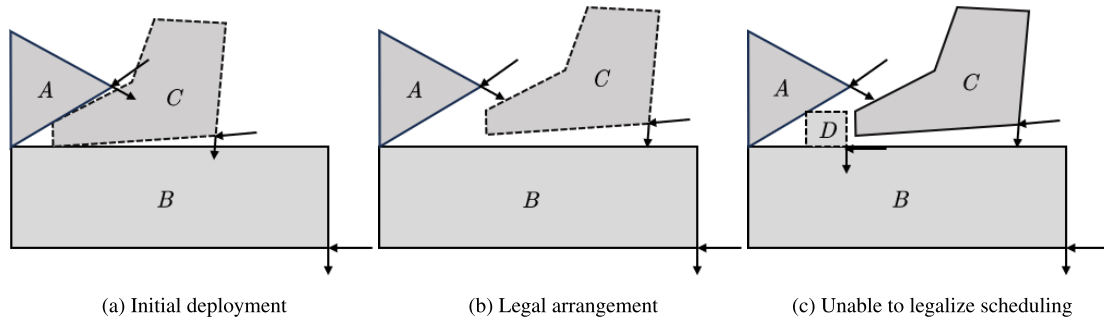


FIGURE 9. The process of lead lines removal overlap.

Algorithm 1 OverlapMinimization

```

Input:  $P$ : the set of all pieces;  $V$ : the set of piece
displacements;
Output: Overlap the overlapped values after removal and
update  $V$ ;
1:  $V_{last} \leftarrow V, V_{best} \leftarrow V$ ;
2:  $maxiter = 3$ ;
3:  $min\_overlap = Overlap(R, V_{last})$ ;
4: while ( $maxiter > 0$ ) do
5:    $V_{last} \leftarrow V$ ;
6:    $overlap\_temp \leftarrow L\_BFGS(O(V_{last}), V_{last})$ ; //this
 $O(V)$  is equation (5).
7:   if  $overlap\_temp == 0$  then
8:      $V_{best} \leftarrow V_{last}$ ;
9:      $min\_overlap \leftarrow 0$ ;
10:    break;
11:  end if
12:  if  $overlap\_temp < min\_overlap$  then
13:     $V \leftarrow V_{last}$ ;
14:     $min\_overlap \leftarrow overlap\_temp$ ;
15:  end if
16:   $maxiter \leftarrow maxiter - 1$ ;
17: end while
18:  $V \leftarrow V_{last}$ 
19: return  $min\_overlap$ ;
    
```

bottom-left algorithm. This algorithm proceeds by traversing the set of points where pieces can potentially be placed, starting from the bottom-left corner and moving towards the top-right corner. It makes legality decisions sequentially until it finds the first legitimate placement point, which is then used as the initial position for the piece. Despite its speed in placement, the bottom-left algorithm often results in a disorderly layout, which can hinder further optimization of the subsequent packing scheme.

To enhance the effectiveness of the bottom-left algorithm, this paper introduces two strategies: the sticking-edge strategy and the piece-insertion strategy. The core of the sticking-edge strategy involves developing a systematic evaluation criterion to guide piece placement decisions. Taking Fig.10 as an illustration, when positioning piece C ,

the algorithm identifies multiple viable placement points and evaluates each one. If the piece employs the lead line overlap removal method outlined in Section IV-C, an additional step ensures no overlap with lead lines of other pieces. Subsequently, the algorithm virtually expands the placed piece C and calculates the fit area at each potential placement point. This includes determining the intersection area with the contours of already placed pieces A and B ($Intersect(C, other)$) as well as the area unique to the piece C and the sheet contour ($Different(C, bin)$) in Fig.10(a) and Fig.10(b). If multiple viable points exist, the algorithm selects the point offering the largest fitting area for placing piece C , as depicted in Fig.10(c). This strategy significantly enhances the compactness of piece arrangements and systematically addresses the fitting relationships between pieces, thereby optimizing conditions for subsequent piece placements.

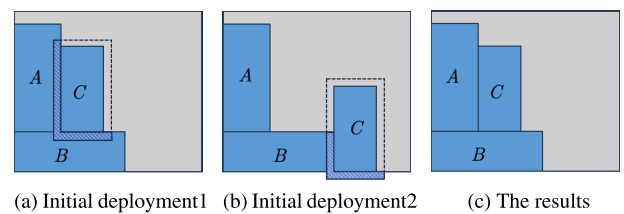


FIGURE 10. The process of sticking-edge strategy.

When dealing with different kinds of piece packing, differences in piece shapes and sizes often create a large number of gaps in the sheet. Although the bottom-left algorithm and the edge-fitting algorithm can efficiently find most of the legal placements, in some cases they may ignore potential locations with only a small amount of overlap due to overly strict legality determination. To address this problem, this paper uses an interpolation algorithm as a supplement when the bottom-left algorithm and the affixed-edge algorithm are unable to find a legitimate solution on the current sheet. This algorithm aims to find the void with the least amount of overlap for piece placement and subsequently calls the overlap removal algorithm for fine-tuning. If the interpolation algorithm is able to find a feasible solution, then we will update the initial solution.

Algorithm 2 Improved Greedy Search Algorithm

Input: P : the set of all pieces; V : the set of piece displacements; R : the set of piece rotations; B : the set of sheets;

Output: update V , R and B ;

```

1: for  $P_i \in P$  do
2:   while  $B_j \in B$  do
3:     Find some legal set of placeable states  $(V_{temp}, R_{temp})$ 
       based on the pieces already placed on sheet  $B_j$ ;
4:     for  $(V_k, R_k) \leftarrow (V_{temp}, R_{temp})$  do
5:       if  $(V_{temp}, R_{temp}) \neq \phi$  then
6:         Legitimate Point Selection Using the sticking-
           edge strategy;
7:       else
8:         Rediscover placement points with the least
           overlap;
9:         if  $OverlapMinimization(P, V_k) > 0$  then
10:          break;
11:        end if
12:      end if
13:       $(V, R) \leftarrow (V_k, R_k)$ ;
14:    end for
15:     $B \leftarrow B_j$ ;
16:  end while
17: end for

```

Conversely, if it is not possible to find a feasible solution, then we skip the currently placed pieces of the same type and continue to try to place another type of piece on the current sheet.

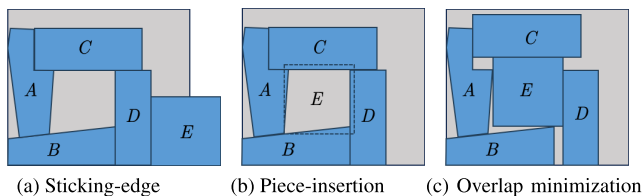


FIGURE 11. The process of piece-insertion strategy.

The piece-insertion algorithm, depicted in Fig.11, is employed when placing piece E . Initially, the sticking-edge strategy is attempted in Fig.11(a), but it results in an illegitimate solution. Consequently, piece E is adjusted to locate a gap that minimizes overlap with already placed pieces in Fig.11(b). Once a suitable placement is identified, the overlap removal algorithm is applied to fine-tune the arrangement, achieving a legal solution in Fig.11(c). Notably, the piece-insertion algorithm is versatile and can accommodate pieces preprocessed using the lead lines removal overlap method. Here, consideration of the lead lines overlap value is integral to determining and addressing overlap during the placement process. This flexibility enables the interpolation algorithm to effectively address various

types of piece packing challenges. The method used to generate the initial solution is shown as Algorithm 2.

B. ITERATIVE COMPRESSION ALGORITHM

After deriving the initial packing solution, there remains significant potential for optimizing the piece layout to maximize sheet space. To achieve this goal, this paper employs an iterative compression strategy. The strategy begins by attempting to enhance piece compactness by compressing the sheet length, which may lead to some pieces overlapping. Subsequently, it applies piece swapping and overlap removal techniques to minimize the overlap value in the obtained solution. If complete elimination of overlap is not feasible through these methods, the algorithm records and updates the set of solutions with the smallest overlap values.

The process then continues with further attempts at piece swapping until a predefined upper limit on the number of swaps is reached. If a feasible arrangement is still unattainable at this stage, it suggests that the compression level might be excessive. In such cases, the compression amount is appropriately reduced, and the process retries to find a feasible scheduling plan. Upon achieving a feasible plan at a specific compression level, the algorithm maintains this compression amount and continues to strive for higher utilization rates. This iterative refinement continues until no further compression of the sheet layout is possible. Any remaining vacant areas on the right side of the sheet are considered potential spaces for future pieces. Fig.12 illustrates the entire iterative optimization process, detailing how progressive compression and piece swapping can effectively enhance sheet utilization by optimizing the arrangement of pieces. Algorithm 3 outlines the pseudo-code of our proposed algorithm.

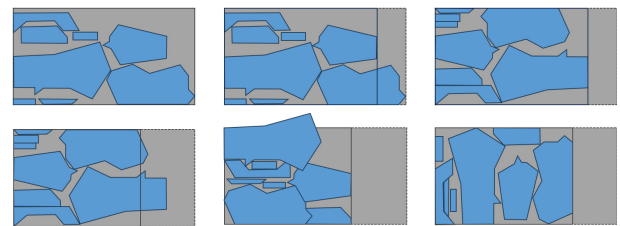


FIGURE 12. Iterative optimization of compression.

VII. RESULTS AND COMPARATIVE ANALYSIS

All algorithms proposed in this paper have been implemented using C++11, and all experiments were conducted on a Windows computer equipped with an Intel(R) Core(TM) i5-10400F CPU running at 2.90 GHz and 16.0 GB of RAM. The effectiveness of these algorithms has been validated through experimental tests conducted on real-world cases within enterprises.

Table 1 gives the basic information of the test case. It mainly contains the length and width of the sheet, the

TABLE 1. The characteristics of the data.

Case	Sheet information		Piece information			Process information		
	Length	Width	Types	Points	Num	Gap	Inlength	Outlength
Base01	7400	1800	18	144.889	112	4.25	9.35	6.95
Base02	6000	1500	11	143.545	252	5	10.4	7.9
Base03	7400	1650	6	13.5	240	5	13.7	8.5
Base04	9200	1800	82	273.341	229	2.87	11.67	9.57
Base05	12800	3000	27	134.259	233	2.5	10.2	8
Base06	7200	2050	51	144.961	286	2.75	6.05	9.45
Base07	8000	2050	46	142.261	222	2.75	11.45	11.65
Base08	6500	1500	47	223.319	246	2.75	7.75	10.05
Base09	8000	2500	9	159.111	255	4.16	6.46	13.76
Base10	6700	1400	38	105.237	211	4.25	5.65	13.65
Base11	7400	1650	40	196.8	295	5	14	9.7
Base12	8000	1000	26	350.962	231	5	14.2	12.5
Base13	6000	1500	10	113.6	362	4.12	10.42	9.82
Base14	9000	2100	29	168.207	446	5	14	12.4
Base15	5400	1800	16	62.4375	468	4.25	10.55	13.15
Base16	6100	1500	53	323.717	368	2.78	8.88	6.88
Base17	6000	1500	19	190	284	2.86	5.96	10.26
Base18	6000	2200	67	287.269	478	2.87	5.07	7.47
avg	7394.44	1805.56	33.06	176.52	289.89	3.79	9.76	10.09

TABLE 2. Result of not considering lead lines and shape expansion method.

Case	Not considering lead lines			Shape expansion method		
	Utilization	Bins num	times(s)	Utilization	Bins num	times(s)
Base01	0.8368	4	501.63	0.8179	4	619.62
Base02	0.8188	3	686.54	0.7881	3	837.09
Base03	0.8973	5	426.65	0.8504	5	572.94
Base04	0.8249	3	752.58	0.7931	4	832.49
Base05	0.7617	2	465.98	0.7292	2	905.96
Base06	0.7888	5	730.09	0.7762	5	905.97
Base07	0.8381	5	827.89	0.8025	5	918.72
Base08	0.8401	4	705.63	0.8167	4	796.90
Base09	0.4992	3	969.92	0.4992	3	971.59
Base10	0.8694	5	783.46	0.8194	5	824.73
Base11	0.8498	6	872.73	0.8256	6	909.69
Base12	0.7158	6	757.54	0.6928	7	797.89
Base13	0.8138	14	821.33	0.7812	15	854.38
Base14	0.7299	4	844.96	0.6959	4	887.90
Base15	0.8083	1	433.35	0.7895	1	545.18
Base16	0.9241	5	548.26	0.8958	5	619.54
Base17	0.8451	2	529.27	0.8136	2	935.41
Base18	0.9035	2	530.66	0.8753	2	628.52
avg	0.8092	4.3889	677.1376	0.7812	4.5556	798.0299

number of types of pieces, the average number of point sets of the piece contour, the number of pieces, the piece gap, the lead-in line length, and the lead-out line length.

Table 2 and Table 3 presents a comparison of three different lead lines processing methods applied to the enterprise dataset. The main statistics in the table are the utilization

TABLE 3. Result of Local deformation method and Lead removal method.

Case	Local deformation method			Lead removal method		
	Utilization	Bins num	times(s)	Utilization	Bins num	times(s)
Base01	0.8341	4	1171.76	0.8318	4	636.62
Base02	0.8188	3	1186.36	0.8188	3	817.07
Base03	0.8973	5	1162.77	0.8973	5	726.68
Base04	0.8249	3	1332.89	0.8249	3	812.36
Base05	0.7524	2	1375.23	0.7559	2	913.99
Base06	0.7793	5	1400.88	0.7826	5	952.03
Base07	0.8347	5	1870.17	0.8435	5	922.48
Base08	0.8351	4	1226.81	0.8416	4	880.04
Base09	0.4992	3	1001.63	0.4992	3	1017.04
Base10	0.8650	5	1386.54	0.8621	5	855.08
Base11	0.8495	6	1230.65	0.8481	6	930.23
Base12	0.7158	6	1252.58	0.7158	6	860.82
Base13	0.8111	14	1165.98	0.8159	14	906.37
Base14	0.7305	4	1330.09	0.7258	4	919.96
Base15	0.8016	1	927.89	0.8033	1	558.83
Base16	0.9214	5	1505.63	0.9248	5	688.95
Base17	0.8377	2	1069.92	0.8325	2	955.75
Base18	0.9012	2	1144.89	0.8998	2	690.01
avg	0.8061	4.3889	1263.4824	0.8069	4.3889	835.7949

TABLE 4. Comparison of algorithm performance evaluation at each stage by lead removal method.

Case	bottom-left algorithm			Initial solution			Iterative compression		
	Utilization	Bins num	Time(s)	Utilization	Bins num	Time(s)	Utilization	Bins num	Time(s)
Base01	0.8031	4	183.6	0.8248	4	204.4	0.8318	4	636.6
Base02	0.7934	3	197.1	0.8122	3	248.4	0.8188	3	817.1
Base03	0.8538	5	119.1	0.8892	5	273.7	0.8973	5	726.7
Base04	0.7853	4	101.2	0.8163	3	134.9	0.8249	3	812.4
Base05	0.7195	2	126.5	0.7485	2	187.4	0.7559	2	914.0
Base06	0.7374	5	269.8	0.7769	5	325.9	0.7826	5	952.0
Base07	0.8047	5	195.4	0.8392	5	291.4	0.8435	5	922.5
Base08	0.8034	4	181.6	0.8346	4	212.5	0.8416	4	880.0
Base09	0.4916	3	122.7	0.4958	3	159.3	0.4992	3	1017.0
Base10	0.8295	5	163.8	0.8573	5	198.6	0.8621	5	855.1
Base11	0.8194	6	177.4	0.8454	6	238.8	0.8481	6	930.2
Base12	0.7072	7	136.1	0.7122	7	172.9	0.7158	6	860.8
Base13	0.7849	15	264.2	0.8062	15	387.5	0.8159	14	906.4
Base14	0.6951	4	185.9	0.7187	4	238.5	0.7258	4	920.0
Base15	0.7749	1	127.3	0.7978	1	165.0	0.8033	1	558.8
Base16	0.9010	5	184.8	0.9129	5	238.5	0.9248	5	689.0
Base17	0.8066	2	101.5	0.8268	2	148.3	0.8325	2	955.8
Base18	0.8682	2	216.4	0.8937	2	287.6	0.8998	2	690.0
avg	0.7766	4.5556	169.6889	0.8005	4.5000	228.5267	0.8069	4.3889	835.7949

of the case results, the number of sheets used and the time the algorithm was used. The statistical results are based on the average of 10 independent runs for each case.

According to the experiments, the lead removal method demonstrates the highest average utilization rate at 80.69%. In contrast, the piece shape expansion method shows the

TABLE 5. Comparison of algorithm performance evaluation at each stage by local deformation.

Case	Bottom-left algorithm			Initial solution			Iterative compression		
	Utilization	Bins num	Time(s)	Utilization	Bins num	Time(s)	Utilization	Bins num	Time(s)
Base 01	0.7852	4	220.5	0.8272	4	269.5	0.8341	4	1171.8
Base 02	0.7896	3	263.1	0.8116	3	375.1	0.8188	3	1186.4
Base 03	0.8582	5	218.5	0.8877	5	389.4	0.8973	5	1162.8
Base 04	0.7866	3	197.2	0.8143	3	273.8	0.8249	3	1332.9
Base 05	0.7271	2	148.5	0.7496	2	187.4	0.7524	2	1375.2
Base 06	0.7365	5	309.2	0.7741	5	408.4	0.7793	5	1400.9
Base 07	0.8012	5	247.1	0.8301	5	327.3	0.8347	5	1870.2
Base 08	0.7924	4	220.7	0.8297	4	289.8	0.8351	4	1226.8
Base 09	0.4911	3	147.4	0.4949	3	201.7	0.4992	3	1001.6
Base 10	0.8269	5	206.4	0.8592	5	274.9	0.8650	5	1386.5
Base 11	0.8149	6	199.7	0.8439	6	288.6	0.8495	6	1230.7
Base 12	0.7011	6	175.2	0.7086	7	243.1	0.7158	6	1252.6
Base 13	0.7821	14	314.8	0.8083	15	477.6	0.8111	14	1166.0
Base 14	0.6974	4	231.3	0.7247	4	364.4	0.7305	4	1330.1
Base 15	0.7758	1	158.8	0.7995	1	221.6	0.8016	1	927.9
Base 16	0.9062	5	220.3	0.9148	5	289.1	0.9214	5	1505.6
Base 17	0.8017	2	144.2	0.8285	2	182.7	0.8377	2	1069.9
Base 18	0.8671	2	257.6	0.8912	2	339.1	0.9012	2	1144.9
avg	0.7745	4.3889	215.5833	0.7999	4.5000	300.1964	0.8061	4.3889	1263.4824

Algorithm 3 Iterative Compression Algorithm

Input: P : the set of all pieces; V : the set of piece displacements; R : the set of piece rotations; B : the set of sheets;

Output: update V , R and B ;

```

1:  $(B, V, R) \leftarrow ImprovedGreedySearchAlgorithm(B, V, R)$ ;
2: for  $B_i \in B$  do
3:    $L_{used} \leftarrow$  Get the used length of the current sheet  $B_i$ ;
4:   while  $compress\_ratio > EPSILON$  do
5:      $L_{used} \leftarrow L_{used} / compress\_ratio$ ;
6:      $(V_{temp}, R_{temp}) \leftarrow$  Through the existing solutions  $(V, R)$  exchange different kinds of pieces, and try different rotation angles;
7:     if  $OverlapMinimization(V_{temp}, R_{temp}) == 0$  then
8:        $V, R \leftarrow V_{temp}, R_{temp}$ ;
9:       continue;
10:    else
11:       $L_{used} \leftarrow L_{used} * (1 + compress\_ratio)$ ;
12:       $compress\_ratio \leftarrow compress\_ratio / 10$ ;
13:    end if
14:  end while
15:   $B \leftarrow B_i$ ;
16: end for

```

lowest average utilization rate of 78.12%. The local deformation of pieces method achieves a moderate effect with an average utilization rate of 80.61%. However, it is important

to note that the piece localized deformation method exhibits longer running times. This is primarily due to increased complexity of the method in solving the NFP of the piece, leading to significant extension of the NFP computation time. Consequently, this impacts the overall efficiency of the algorithm. To evaluate the effectiveness of various mechanisms in the algorithm presented in this paper, we conducted tests using the bottom-left algorithm, the improved greedy search algorithm, and the iterative compression algorithm on the enterprise case study. Throughout the testing process, all three methods employed the same lead lines preprocessing.

The test results of the three algorithms employing local deformation of pieces are summarized in Table 4. The main statistics in the table are the utilization of the case results, the number of sheets used and the time the algorithm was used. From the table, it is evident that the bottom-left algorithm achieves an average utilization of 77.66%, the improved greedy search algorithm achieves 80.05%, and the iterative compression algorithm achieves 80.69%. Compared to the bottom-left algorithm, the improved greedy search algorithm shows an average improvement of 2.39%, while the iterative compression algorithm demonstrates an average improvement of 3.03%. Furthermore, Table 5 presents the test results of the same algorithms using the lead removal method: the bottom-left algorithm achieves an average utilization of 77.45%, the improved greedy search algorithm achieves 79.99%, and the combined use of the improved greedy search algorithm with the compression algorithm

achieves 80.61%. These results indicate that the iterative compression algorithm performs best under both lead lines preprocessing methods. Moreover, the improved greedy search algorithm exhibits significant improvement over the bottom-left algorithm, highlighting the effectiveness of the sticking-edge, piece-insertion, and compression algorithms proposed in this paper.

VIII. CONCLUSION AND FUTURE SCOPE

In this paper, a comprehensive approach is presented for addressing the 2D irregular packing problem with consideration of lead lines. The proposed solution includes three preprocessing methods for pieces and two heuristic packing algorithms. Initially, the piece preprocessing methods effectively convert complex lead line constraints into simplified non-overlapping constraints among pieces, thereby facilitating the management of the packing problem. Subsequently, pieces are densely packed onto the sheet using heuristic strategies such as sticking-edge, piece-insertion, and compression, aimed at maximizing sheet space utilization. To validate the efficacy of this approach, a real-world cases within enterprises case study was conducted. The results demonstrate that local piece deformation and lead removal method during preprocessing significantly mitigate the impact of lead lines on piece arrangement. Furthermore, the overlap minimization of algorithmic modules and their impact on final packing outcomes were assessed to confirm the effectiveness of the strategy. Experimental findings indicate a notable enhancement in raw material sheet utilization, thereby validating the efficacy of the proposed piece preprocessing and packing methods. This study contributes novel insights and methodologies to address the irregular packing problem involving lead lines in the sheet metal industry.

Future work will focus on addressing these limitations by exploring solutions for packing lead-in lines with circular arc shapes and investigating methods to enhance the speed and efficiency of critical polygon solving, thereby optimizing the overall performance of the packing algorithm. In addition, some digital twin technology [19] can be introduced to digitize the packing process, machine learning [20] and learning network [21] can be used to detect possible problems in the cutting process.

REFERENCES

- [1] A. A. S. Leao, F. M. B. Toledo, J. F. Oliveira, M. A. Carravilla, and R. Alvarez-Valdés, "Irregular packing problems: A review of mathematical models," *Eur. J. Oper. Res.*, vol. 282, no. 3, pp. 803–822, May 2020.
- [2] S. Cai, J. Deng, L. H. Lee, E. P. Chew, and H. Li, "Heuristics for the two-dimensional irregular bin packing problem with limited rotations," *Comput. Oper. Res.*, vol. 160, Dec. 2023, Art. no. 106398.
- [3] E. López-Camacho, G. Ochoa, H. Terashima-Marín, 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.
- [4] Q. Liu, J. Zeng, H. Zhang, and L. Wei, "A heuristic for the two-dimensional irregular bin packing problem with limited rotations," in *Proc. 33rd Int. Conf. Ind., Eng. Appl. Intell. Syst.*, Kitakyushu, Japan. Berlin, Germany: Springer, Sep. 2020, pp. 268–279.

- [5] H. Zhang, Q. Liu, L. Wei, J. Zeng, J. Leng, and D. Yan, "An iteratively doubling local search for the two-dimensional irregular bin packing problem with limited rotations," *Comput. Oper. Res.*, vol. 137, Jan. 2022, Art. no. 105550.
- [6] H. Okano, "A scanline-based algorithm for the 2D free-form bin packing problem," *J. Oper. Res. Soc. Jpn.*, vol. 45, no. 2, pp. 145–161, 2002.
- [7] H. Terashima-Marín, P. Ross, C. J. Fariás-Zárate, E. López-Camacho, and M. Valenzuela-Rendón, "Generalized hyper-heuristics for solving 2D regular and irregular packing problems," *Ann. Oper. Res.*, vol. 179, no. 1, pp. 369–392, Sep. 2010.
- [8] J. C. Gomez and H. Terashima-Marín, "Evolutionary hyper-heuristics for tackling bi-objective 2D bin packing problems," *Genetic Program. Evolvable Mach.*, vol. 19, nos. 1–2, pp. 151–181, Jun. 2018.
- [9] E. López-Camacho, H. Terashima-Marín, P. Ross, and G. Ochoa, "A unified hyper-heuristic framework for solving bin packing problems," *Exp. Syst. Appl.*, vol. 41, no. 15, pp. 6876–6889, Nov. 2014.
- [10] R. P. Abeysooriya, J. A. Bennell, and A. Martínez-Sykora, "Jostle heuristics for the 2D-irregular shapes bin packing problems with free rotation," *Int. J. Prod. Econ.*, vol. 195, pp. 12–26, Jan. 2018.
- [11] S. Yao, C. Tang, H. Zhang, S. Wu, L. Wei, and Q. Liu, "An iteratively doubling binary search for the two-dimensional irregular multiple-size bin packing problem raised in the steel industry," *Comput. Oper. Res.*, vol. 162, Feb. 2024, Art. no. 106476.
- [12] S. Yao, H. Zhang, Q. Liu, J. Leng, and L. Wei, "Combinatorial benders' decomposition for the constrained two-dimensional non-guillotine cutting problem with defects," *Int. J. Prod. Res.*, pp. 1–27, Apr. 2024.
- [13] W. Han, J. A. Bennell, X. Zhao, and X. Song, "Construction heuristics for two-dimensional irregular shape bin packing with guillotine constraints," *Eur. J. Oper. Res.*, vol. 230, no. 3, pp. 495–504, Nov. 2013.
- [14] A. Martínez-Sykora, R. Alvarez-Valdes, J. A. Bennell, R. Ruiz, and J. M. Tamarit, "Matheuristics for the irregular bin packing problem with free rotations," *Eur. J. Oper. Res.*, vol. 258, no. 2, pp. 440–455, Apr. 2017.
- [15] J. A. Bennell, M. Cabo, and A. Martínez-Sykora, "A beam search approach to solve the convex irregular bin packing problem with guillotine cuts," *Eur. J. Oper. Res.*, vol. 270, no. 1, pp. 89–102, Oct. 2018.
- [16] P. K. Agarwal, L. J. Guibas, S. Har-Peled, A. Rabinovitch, and M. Sharir, "Penetration depth of two convex polytopes in 3D," *Nord. J. Comput.*, vol. 7, no. 3, pp. 227–240, 2000.
- [17] M. A. Abramson, G. D. Kent, and G. W. Smith, "Penetration depth between two convex polyhedra: An efficient stochastic global optimization approach," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 12, pp. 4267–4273, Dec. 2022.
- [18] Y. J. Kim, M. C. Lin, and D. Manocha, "Incremental penetration depth estimation between convex polytopes using dual-space expansion," *IEEE Trans. Vis. Comput. Graphics*, vol. 10, no. 2, pp. 152–163, Apr. 2004.
- [19] J. Xia, R. Huang, J. Li, Z. Chen, and W. Li, "Digital twin-assisted fault diagnosis of rotating machinery without measured fault data," *IEEE Trans. Instrum. Meas.*, 2024.
- [20] S. Ma, J. Leng, Z. Chen, B. Li, X. Li, D. Zhang, W. Li, and Q. Liu, "A novel weakly supervised adversarial network for thermal error modeling of electric spindles with scarce samples," *Exp. Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122065.
- [21] Z. Chen, J. Xia, J. Li, J. Chen, R. Huang, G. Jin, and W. Li, "Generalized open-set domain adaptation in mechanical fault diagnosis using multiple metric weighting learning network," *Adv. Eng. Informat.*, vol. 57, Aug. 2023, Art. no. 102033.



CHAO TANG received the B.S. degree in mechanical engineering from Hunan Institute of Science and Technology, Yueyang, China, in 2018. He is currently pursuing the M.S. degree in mechanical engineering with Guangdong University of Technology, Guangzhou, China. His research interests include irregular packing, intelligent manufacturing, computational intelligence, and intelligent optimization.



SHAOWEN YAO received the B.S. degree from Jinggangshan University, Ji'an, China, in 2020. He is currently pursuing the Ph.D. degree with Guangdong University of Technology, Guangzhou, China. His current research interests include intelligent manufacturing and optimization algorithms.



SHIGANG ZHANG received the B.S. degree in mechanical engineering from Hunan Institute of Science and Technology, Yueyang, China, in 2018. He is currently pursuing the M.S. degree in mechanical engineering with Guangdong University of Technology, Guangzhou, China. His research interests include irregular packing, intelligent manufacturing, computational intelligence, and intelligent optimization.



LIMEI LU was born in 1971. She received the bachelor's degree in information management and information systems from Hangzhou University of Electronic Science and Technology, in 2010. She is currently a Senior Engineer with Shanghai Aerospace Power Technology Research Institute, China. She has published more than ten research results and won the third prize for management innovation achievements in China's defense technology and industrial enterprises, in 2021. Her research interests include quality management system construction, design of solid engine propellant production testing systems, and supply chain management.



LIJUN WEI received the B.S. and M.S. degrees from Xiamen University, Xiamen, China, and the Ph.D. degree in management sciences from the City University of Hong Kong, Hong Kong, in 2013. He is currently a Professor with Guangdong University of Technology, Guangzhou, China. His current research interests include intelligent algorithm, intelligent manufacturing, and intelligent transport systems.

...