ENGINEERING Information Technology & Electronic Engineering
https://www.academax.com/EITEE; https://www.engineering.org.cn
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11344065
E-mail: eng.itee@zju.edu.cn

Engineering Journals
Available on IEEE Xplore®

Research Article

https://doi.org/10.1631/ENG.ITEE.2025.0080

Hierarchical algorithm for large-scale irregular packing problems

Xiao LIU✉

School of Civil Engineering & Transportation, South China University of Technology, Guangzhou 510640, China

Abstract: To address the challenge of large-scale packing problems, this paper proposes a novel hierarchical algorithm based on the geometrical classification of parts. The algorithm begins by classifying parts into three levels based on their area and fullness and then applies distinct packing strategies to each category. An innovative "shape matching" method is introduced, which, together with the "box stacking" (for rectangular parts) and "gravity packing," forms a comprehensive hierarchical packing system. Level-1 comprises large rectangular parts, which are arranged using the box stacking algorithm. By aligning the corner points of the parts' bounding boxes, this method avoids the hooking issue commonly encountered in gravity packing. Level-2 includes both large, irregular parts and medium-sized parts. They are first processed using the shape matching algorithm, where rotation and translation are applied to achieve contour complementarity. The quality of the match is evaluated using the shape matching coefficient (SMC). If the SMC fails to reach the preset quality threshold, the system switches to box stacking (for large, irregular parts) or gravity packing (for medium-sized parts). Level-3 comprises the remaining smaller parts and those that failed to pack in the previous two levels. For these parts, shape matching is attempted first, and the system resorts to gravity packing in case of failure. The experimental and comparative results demonstrate that the proposed hierarchical algorithm achieves higher material utilization than the traditional gravity packing algorithm. This improvement is facilitated by the box stacking and shape matching strategies, which promote a more orderly and compact arrangement of parts.

Key words: Large-scale packing; Hierarchical algorithm; Box stacking; Shape matching; Gravity packing; Principle of minimum potential energy

1 Introduction

Packing problems involve the optimal arrangement of parts of varying shapes and sizes on raw materials, such as metal sheets, fabrics, and wood, to maximize material utilization while satisfying process constraints, including orientation and gap requirements (Francescatto and Júnior, 2025). This process is widely employed in industries such as sheet metal processing, garment manufacturing, and furniture production, and directly affects both material costs and production efficiency.

The trend in manufacturing is increasingly leaning toward ever-larger scales. Taking the shipbuilding industry as an example, the larger a vessel's deadweight tonnage is, the lower its unit manufacturing and cargo transportation costs. This trend toward large-scale manufacturing presents three primary challenges for packing problems: (1) a large quantity of parts; (2) irregular part geometries; (3) the use of multiple sheets (raw material plate). However, no existing studies have effectively tackled all three issues in an integrated manner. While Zhou et al. (2024) and Lai et al. (2025) focused on the large-scale packing problem of regular parts (with the number of circles reaching 1000), research on the packing of irregularly shaped parts (Stoyan and Pankratov, 1999; Cheng and Rao, 2000; Costa et al., 2009; Al Theeb et al., 2021; Liu YL and Zheng, 2025) is limited to a maximum of 450 parts and one single sheet. Guo et al. (2025) investigated the packing of irregular parts on multiple sheets, but their study only considered a maximum of 130 parts.

This paper proposes a new packing algorithm that can pack 1400 irregular parts across multiple sheets while maintaining the computation time suitable for engineering applications. The algorithm classifies parts by area and fullness and employs a combination of "box stacking," "shape matching," and "gravity packing" strategies. Among these, box stacking is essentially a form of rectangular packing, gravity packing refers to an algorithm developed over the past decade, and shape matching is a novel method introduced in the present work. Before presenting the proposed approach, it is helpful to briefly review traditional algorithms for rectangular and irregular part packing.

In certain engineering applications, the area of rectangular parts constitutes a significant proportion of the total area. For instance, in shipbuilding, the presence of a parallel mid-body results in a large number of parts that are approximately rectangular in shape (hereinafter referred to as "box-like parts"), with the remaining parts

✉ Xiao LIU, liuxiao@scut.edu.cn
Xiao LIU, https://orcid.org/0000-0003-2975-4749
CLC number: TP301.6

Received: Oct. 12, 2025; Revision accepted: Feb. 4, 2026;
Crosschecked: Feb. 12, 2026; Published online: Mar. 16, 2026

© The Author 2026. Published by Zhejiang University Press Co., Ltd.
This is an open access article distributed under the terms of the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

being irregular. The area of these box-like parts can account for up to 58.0% of the total area encompassing all parts.

The placement of rectangular parts is straightforward and, due to the absence of overlapping detection, is significantly faster than that of irregular parts. Common rectangular layout heuristic algorithms include the below-left (BL) heuristic algorithm (Baker et al., 1980), corner-occupying action (Chen and Huang, 2007), and four corners' heuristic algorithm (Binkley and Hagiwara, 2007). Among these, the BL heuristic algorithm is the earliest and most widely used algorithm. Its principle is as follows: a rectangular part starts at the top-right corner of the sheet, moves downward as far as possible and then leftward until it can no longer move, and repeats this sequence until it can move no further.

Because of its simplicity and strong performance, the BL strategy has been adopted in subsequent research. However, its inherent shortcomings have prompted the development of numerous improved BL-based heuristics algorithm. A notable example is the down priority BL strategy (DPBL) method proposed by Liu DQ and Teng (1999), which was designed to overcome the specific limitations of the original algorithm. The DPBL holds that downward translational movement takes precedence over leftward movement, which is more consistent with the motion law of objects in a gravitational field and reflects gravity's influence on object motion. In addition to the considerations addressed by the DPBL, another common issue with the BL algorithm is its tendency to leave unused "holes" in the sheet. To solve this problem, the bottom left filling (BLF) algorithm was developed (Chazelle, 1983). This algorithm records the position points formed by the previously packed parts. For each new part to be placed, it visits these key points (i.e., aligns its bottom-left corner with the position point) and calculates the height of its center of gravity at each location. By comparing these heights, the position corresponding to the lowest center of gravity is selected as the optimal placement. This strategy encourages parts to settle into lower, more stable positions, thereby improving the utilization of available space.

While the above methods focus primarily on rectangular parts, the challenge of packing irregular parts has also been extensively studied, often building upon principles derived from rectangular packing algorithms. For instance, Hopper (2000) replaced irregular parts with their bounding boxes, thereby converting the irregular packing problem into a rectangular one. However, this approach evidently fails to fully use the gaps between irregular parts. After decades of development, two main research directions have emerged in the field of irregular part packing: no-fit polygon (NFP)-based packing and gravity packing.

NFP was employed to generate non-overlapping regions, in which an unpacked part can be placed in contact with the already packed parts. Since its application to packing problems by Albano and Sapuppo (1980), NFP has been widely adopted by researchers. The key advantage of this method is its capacity to exhaustively locate all optimal positioning points around packed parts. Nevertheless, this approach is subject to constraints. Specifically, any rotation of the part necessitates the recalculation of NFP, and the algorithm for generating NFPs for concave polygons is notably complex and computationally demanding.

Dowsland et al. (1998) proposed a jostling algorithm that treats parts like granular material in a bottle. Shaking the "bottle" continuously can make the "granular stuff" more compact. Despite this, they readily acknowledged that the underlying mechanism by which lateral shaking improves packing density was unclear.

Liu HY and He (2006) proposed a method to identify the optimal packing position by selecting the location with the lowest center of gravity using the NFP. Their procedure is as follows: first, the part's center of gravity is set as the reference point; then, an internal NFP is generated for each possible rotation angle of the part; finally, the position with the lowest center of gravity is identified among these NFPs. The coordinates and rotation angle corresponding to this position define the optimal packing attitude for the part.

Lee et al. (2008) introduced an algorithm that places a set of equally spaced candidate points in the sheet. For each point, the part is translated to that location and rotated around it while continuously evaluating the variation of its centroid (x, y). The final packing attitude is determined by the position and angle that minimize the distance from the part's center to the origin of the coordinates.

As shown in the three papers above, gravity plays an essential role in the packing process. Inspired by this insight, Liu X and Ye (2011) used the principle of minimum potential energy to explain the physical meaning of part movement in packing problems and proposed a new irregular part packing algorithm—a heuristic algorithm based on the principle of minimum total potential energy (HAPE). The principle of minimum potential energy is a concept in structural mechanics that explains the variation of energy $\Pi=U+V$ in a structure during loading and deformation (U is the elastic potential energy and V is the positional potential energy): a structural system always tends to keep its energy as low as possible. Given that the parts do not deform during the packing process, their elastic potential energy can be neglected; thus, only $V=Gy$ needs to be considered. Given that G is a constant, the system seeks to minimize the part's y value to achieve the lowest possible energy. HAPE generates a large set of packing points across the plate. A part explores these points by being sequentially translated and rotated at each location while checking for collisions with the sheet boundary or the already packed parts.

While HAPE achieves competitive packing results as a gravity-driven method, it prioritizes satisfying physical principles over enhancing the material utilization rate. In contrast, this paper proposes a novel hierarchical algorithm with the primary goal of maximizing material utilization rates for large-scale packing problems. It classifies parts into three distinct groups based on size and shape fullness. The first group is arranged using the box stacking method; the second employs a shape matching approach, with either box stacking or gravity packing as a fallback; the third uses either shape matching or gravity packing. By applying appropriate packing strategies to each part type, this method prioritizes layout regularity and packing tightness, thereby achieving higher material utilization than physics-first approaches such as HAPE. The effectiveness of the proposed algorithm is validated through two experimental studies. In recent years, Guerriero and Saccomanno (2023) also proposed a hierarchical algorithm. However, this hierarchy primarily addressed heuristic algorithms rather than the dimensions and shapes of the parts. Furthermore, the maximum number of parts handled in the experiments discussed in their study was 60, and rotation or flipping of the parts was not permitted.

2 Hierarchical packing algorithm

Before elaborating on the hierarchical packing algorithm, it is necessary to define the size and fullness of parts.

2.1 Definitions

1. Part size

Parts are defined by size according to the area of their outer domain (excluding inner holes), as shown in Table 1. It should be noted that part sizes are not absolute; the standard for large, medium, and small parts shown in Table 1 is specific to the field of shipbuilding engineering, such as container ships with a gross tonnage of around 100 000 tons. Users in other engineering fields should select an appropriate area threshold based on their own characteristics or calculate it using Eq. (1).

$$\begin{cases} A_0=\left(\dfrac{W}{12}\right)^2, \\ A_1=\left(\dfrac{W}{4}\right)^2, \end{cases} \tag{1}$$

where W is the width of the sheet.

Table 1 Part size (outer domain area)

Range of A (m^2)	Part size
$A \geq A_1$	Large
$A_0 \leq A < A_1$	Medium
$A < A_0$	Small

A represents the outer domain area, A_1=0.5 m^2, and A_0=0.05 m^2

2. Block coefficient

Parts can be categorized according to a property referred to as their fullness (Fig. 1), which is defined by the block coefficient C_b:

$$C_b=A_o/A_b, \tag{2}$$

where A_o is the area of the part's outer domain, and A_b is the area of the part's bounding box.

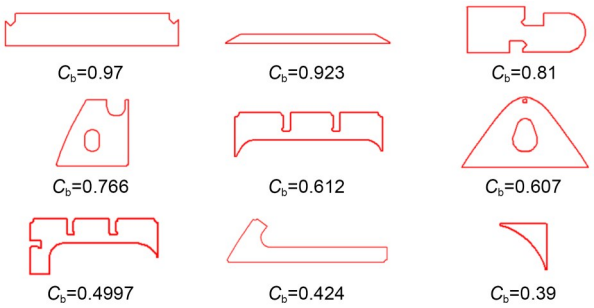

Fig. 1 Block coefficients of various parts

2.2 Gravity packing

The heuristic packing algorithm HAPE uses the principle of minimum potential energy in structural mechanics: during the packing process, parts always try to lower the height of their center of gravity through translation or rotation (Fig. 2). Since part packing is controlled by gravity, we might also call the HAPE algorithm "gravity packing."

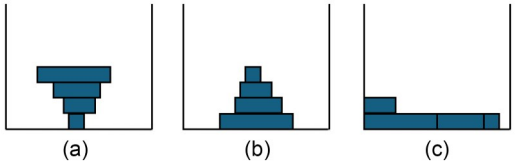

Fig. 2 Gravity packing (gravity directed vertically downward): (a) unstable (high center of gravity); (b) relatively stable (lower center of gravity); (c) the most stable (lowest center of gravity)

As illustrated in Fig. 3, during gravity packing, a part must visit each packing point on the sheet and rotate by specified angles. After eliminating infeasible packing attitudes, the one with the lowest center-of-gravity height is selected as the final attitude. For the specific implementation method, please refer to Liu X and Ye (2011).

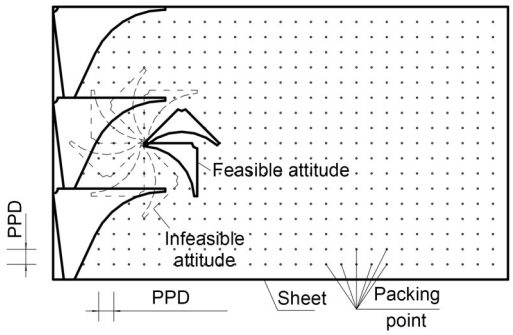

Fig. 3 Packing points and attitudes (gravity directed horizontally leftward). PPD: packing point distance

2.3 Shape matching

Shape matching refers to the process whereby two parts with similar shapes achieve shape complementarity through rotation and sliding. As shown in Fig. 4, the green part A has been packed and fixed to the sheet, and the red part B is ready to be packed. After rotating 180° and sliding downward, part B is fitted together with part A.

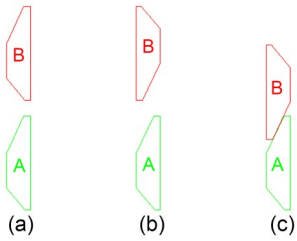

Fig. 4 Shape matching: (a) parts with similar shapes; (b) part B rotated by 180°; (c) part B slides downward to be fitted together with part A

While gravity packing follows the physical laws governing part movement, it does not prioritize enhanced material utilization in all cases. As depicted in Fig. 5, part A has already been packed, while part B is awaiting packing. According to the principle of minimum potential energy, part B will adopt the packing attitude

shown in Fig. 5a to lower its center of gravity. Conversely, if the shape matching algorithm is used, part B will be packed as illustrated in Fig. 5b. Although this results in a higher center of gravity for part B, it frees up more space in the bottom-right corner of the sheet, thereby creating more favorable conditions for packing a third part C, as demonstrated in Fig. 6b.

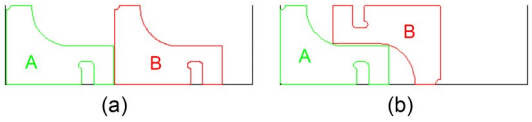

Fig. 5 Two-part packing (gravity directed downward): (a) gravity packing; (b) shape matching

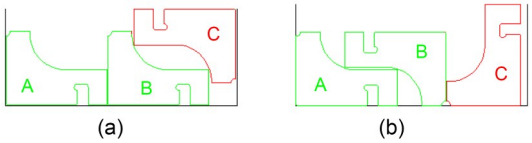

Fig. 6 Three-part packing (gravity directed downward): (a) gravity packing; (b) shape matching

The following provides a detailed explanation of the relevant concepts and workflow for shape matching.

2.3.1 Definitions related to shape matching

1. Shape matching triggering criteria

Shape matching can be triggered by a similarity in one dimension (length or width) of the bounding boxes, defined by a relative deviation of <1%. As shown in Fig. 7, the process can still proceed for parts A and B because their widths match exactly, even with a significant difference in length.

Fig. 7 Part shape matching

2. Shape matching coefficient

As illustrated in Fig. 7, assume that part A has already been packed, and part B is ready to be packed. To compare the effect of shape matching between parts A and B, the shape matching coefficient (SMC) is designed as

$$\text{SMC}=\text{detArea}/\text{blankArea}\times 100\%, \tag{3}$$

where detArea represents the reduction in the bounding box area, which equals the sum of the respective bounding box areas of parts A and B minus the bounding box area of part AB assembly after shape matching; blankArea represents the total blank areas of parts A and B. Blank area represents the blank area of a part, i.e., $A_b(1-C_b)$, where C_b is the block coefficient and A_b is the area of the part's bounding box, as shown in Eq. (2).

As depicted in Fig. 8, the blank area refers to the total area of the two unshaded regions (marked with arrows), which is the portion remaining after subtracting the shaded area from the part's bounding box.

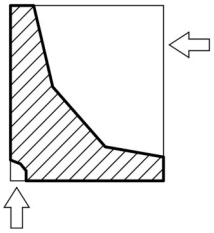

Fig. 8 Part bounding box and its blank area

3. Shape matching quality

A larger SMC indicates better shape matching quality. The rationale behind Eq. (3) is explained below.

As shown in Fig. 9, after shape matching of parts A and B, the area of the bounding box of their combination is smaller than the sum of their individual bounding box areas. The difference between them is the detArea, which corresponds to the area of the shaded region in Fig. 9. Clearly, a larger detArea indicates better shape matching quality.

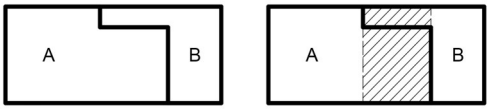

Fig. 9 Reduction in bounding box area

As shown in Fig. 10, the blank areas of part A align perfectly with the protruding areas of part B, and vice versa, enabling a perfect shape match between them. The shaded detArea in Fig. 9 equals the sum of the blank areas of parts A and B in Fig. 10. Substituting this value into Eq. (3) yields that SMC=100%, which is consistent with subjective visual perception. A similar ideal compatibility can be observed in Fig. 7.

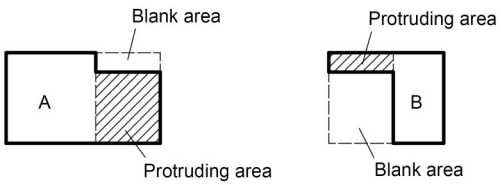

Fig. 10 Blank area and protruding area of parts

As shown in Fig. 11, the blank areas of parts A and B are 36 mm^2 and 25 mm^2, respectively, and their blankArea=61 mm^2. As shown in Fig. 12, after shape matching, the reduction in the parts' bounding box area detArea=6×8=48 mm^2. Substituting them into Eq. (3) gives SMC=48/61=78.7%, which is also consistent with the subjective visual perception.

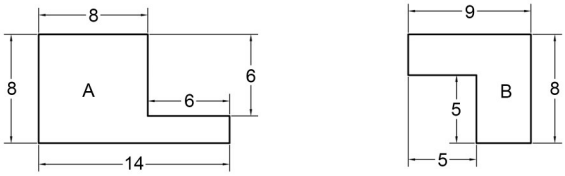

Fig. 11 Part dimensions (unit: mm)

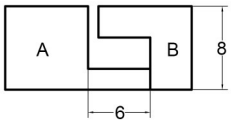

Fig. 12 Dimensions of the overlapping region of the part bounding boxes (unit: mm)

2.3.2 Shape matching algorithm

As illustrated in Fig. 13, let part A be a packed part and part B an unpacked part. Once they meet the shape-matching triggering criteria, part B will perform shape matching with part A via sliding and rotation.

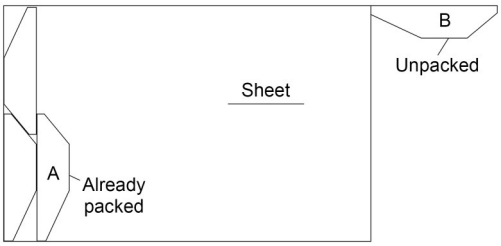

Fig. 13 Two parts trigger shape matching

For the sake of discussion, the shape matching of part B to part A is temporarily assumed to be achieved solely through sliding contact—a process detailed in the algorithm by Liu X and Ye (2011).

1. Shape matching via sliding contact

Let part A be packed, and part B be ready to slide for shape matching with part A.

(1) Shape matching from the right side of part A

(a) As shown in Fig. 14, move part B to align the bottom-left corner of its bounding box with the bottom-right corner of part A's bounding box.

Fig. 14 Part B's bottom-left corner coincides with part A's bottom-right corner

(b) As depicted in Fig. 15, part B slides leftward to contact part A.

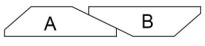

Fig. 15 After contact with part A

(c) If part B does not conflict with other parts or the sheet, calculate the SMC with reference to Eq. (3) and record part B's packing attitude att_R (att_R stores the positioning point and rotation angle required for part B). Here, att_R refers to the packing attitude obtained when part B slides leftward from the right side of the already-packed part A.

(2) Shape matching from the top of part A

(a) As shown in Fig. 16, move part B to align the bottom-left corner of its bounding box with the top-left corner of part A's bounding box.

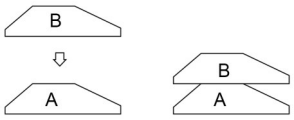

Fig. 16 Part B slides downward to contact with part A

(b) Move part B downward to contact with part A.

(c) If part B does not conflict with other parts or the sheet, calculate the SMC and record its attitude att_T. Here, att_T refers to the packing attitude obtained when part B slides downward from the top of the already-packed part A.

(3) Shape matching from the bottom of part A

(a) As illustrated in Fig. 17, move part B to align the top-left corner of its bounding box with the bottom-left corner of part A's bounding box.

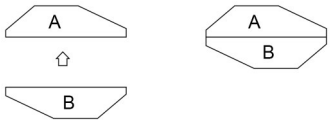

Fig. 17 Part B slides upward to contact with part A

(b) Move part B upward to contact with part A.

(c) If part B does not conflict with other parts or the sheet, calculate the SMC and record its attitude att_B. Here, att_B refers to the packing attitude obtained when part B slides upward from the bottom of the already-packed part A.

2. Shape matching via both rotation and sliding

Part B in Fig. 18 can be placed in four orientations. Once an orientation is determined, we perform the previously described shape matching via sliding contact.

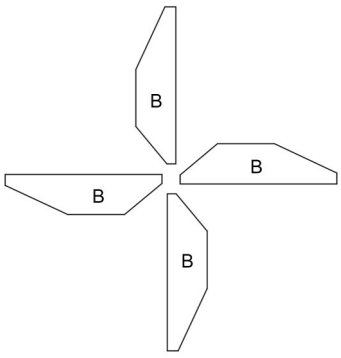

Fig. 18 Four rotational attitudes of part B

The pseudocode in Algorithm 1 illustrates the complete shape matching process. Special attention should be paid to SMC_min, which serves as the threshold: if the SMC meets or exceeds this threshold, shape matching is performed; otherwise, the part is packed using the gravity packing algorithm (HAPE). In this way, the algorithm actively seeks a better material usage rate rather than merely conforming to physical constraints.

As shown in Fig. 19, shape matching produces a highly compact layout for all parts. The results in Fig. 20, obtained via gravity packing, exhibit both poor regularity and low material utilization.

Algorithm 1 Shape matching algorithm

Input:

PartA: packed reference part (contains geometric information: positioning point, vertex coordinates, etc.)

PartB: to-be-packed target part (contains geometric information: positioning point, vertex coordinates, etc.)

SMC_min: minimum threshold of the shape matching coefficient (a preset value)

Output:

att_opt: the optimal positioning attitude of PartB (positioning point coordinates+rotation angle)

pack_strategy: the final packing strategy for PartB (shape matching based packing/gravity packing algorithm HAPE)

SMC_opt: the optimal shape matching coefficient

```
// Step 1: multi-angle and multi-direction shape matching to determine the
// optimal attitude
Initialize att_R[4], att_T[4], att_B[4]  // Initialize arrays to store attitudes
   // of PartB
Initialize SMC_R[4], SMC_T[4], SMC_B[4] // Initialize arrays to store
   // SMC values
SMC_opt=−1.0  // Initialize the optimal SMC to a minimal value
att_opt = null  // Initialize the optimal attitude as null
// Iterate over four rotation angles (0°, 90°, 180°, 270°)
for i=0 to 3 do
      // Rotate PartB around its positioning point by i×π/2 rad
      rotated_PartB=Rotate(PartB, PartB.positioning_point, i×π/2)
      // (I) Perform shape matching via sliding contact from the right side
      // of PartA, and calculate SMC using Eq. (3)
      SMC_R[i]=ShapeMatching(rotated_PartB, PartA, "right")
      att_R[i]=(rotated_PartB.positioning_point, i×π/2)  // Record the
      // attitude (positioning point+rotation angle)
      // (II) Perform shape matching via sliding contact from the top of PartA
      SMC_T[i]=ShapeMatching(rotated_PartB, PartA, "top")
      att_T[i]=(rotated_PartB.positioning_point, i×π/2)
      // (III) Perform shape matching via sliding contact from the bottom of
      // PartA
      SMC_B[i]=ShapeMatching(rotated_PartB, PartA, "bottom")
      att_B[i]=(rotated_PartB.positioning_point, i×π/2)
end for
   // Screen the optimal attitude with the maximum SMC from all candidates
   // Integrate all attitudes and their corresponding SMC values
all_attitudes=[att_R[0..3], att_T[0..3], att_B[0..3]]
all_SMC=[SMC_R[0..3], SMC_T[0..3], SMC_B[0..3]]
   // Traverse all candidates to find the optimal solution
for j=0 to LENGTH(all_SMC)−1 do
   if all_SMC[j]>SMC_opt then
      SMC_opt=all_SMC[j]
      att_opt=all_attitudes[j]
   end if
end for
// Step 2: determine the final packing strategy based on the optimal SMC
SMC_min = 0.25  // Set the minimum threshold for shape matching
if SMC_opt>SMC_min then
   pack_strategy="Shape Matching based packing (adopt att_opt)"
   Pack(PartB, att_opt)  // Pack PartB with the optimal attitude
else
   pack_strategy="Gravity Packing Algorithm HAPE"
      Pack(PartB, HAPE_Algorithm())  // Pack PartB using the HAPE
      // algorithm
end if
```

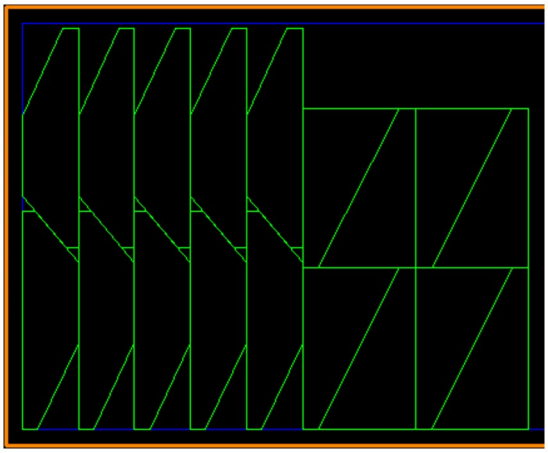

Fig. 19 Shape matching (packing density=80.89%)

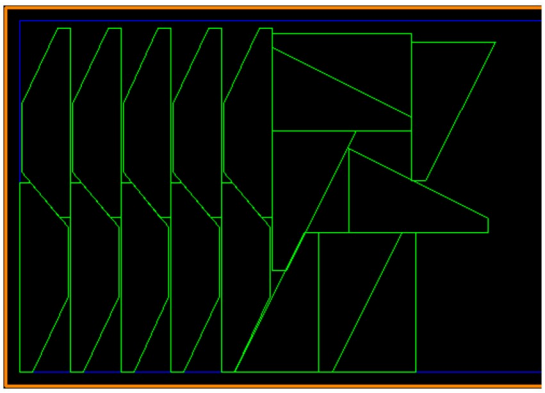

Fig. 20 Gravity packing (packing density=74.16%)

2.4 Box stacking

Before detailing the box stacking algorithm, it is essential to understand its motivation. When packing parts with a high block coefficient (C_b>0.9), gravity packing methods often lead to localized interlocking or hooking between adjacent parts. As a result, the layout becomes disorderly, leaving multiple small, underutilized segments across the sheet. This effect is illustrated in Fig. 21: under a leftward horizontal gravitational field, the red-highlighted region minimizes the x coordinate of its center of gravity (following

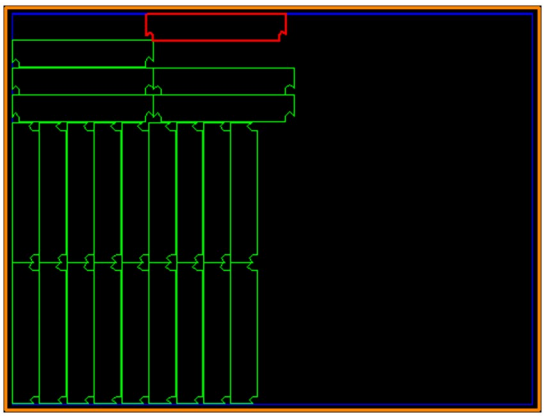

Fig. 21 Hooking of box parts caused by gravity packing (assuming that gravity is directed horizontally leftward)

the principle of minimum potential energy), causing it to hook with the green region at the bottom-left. Such local optimization at the expense of global efficiency is analogous to the traffic congestion caused by a few drivers cutting in line (Fig. 22). Conversely, Fig. 23 shows how smooth flow is restored when rules are followed, which is the core idea behind the box stacking algorithm.

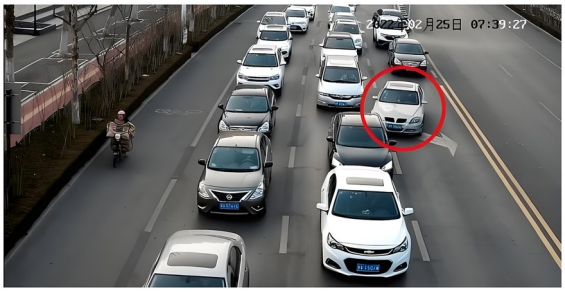

Fig. 22 Traffic congestion caused by cutting in line

Fig. 23 Compliance with traffic rules

As shown in Fig. 24, if a part possesses both a large area and a high block coefficient (C_b>0.9), the box stacking algorithm should be prioritized. This algorithm is essentially a rectangular packing algorithm: the bounding box of the part to be packed is aligned with one of the corners of the bounding box of an already-packed part. In Fig. 25, the bottom-right corner of part A's bounding box coincides with the bottom-left corner of part B's bounding box; the top-left corner of part B's bounding box coincides with the bottom-left corner of part C's bounding box.

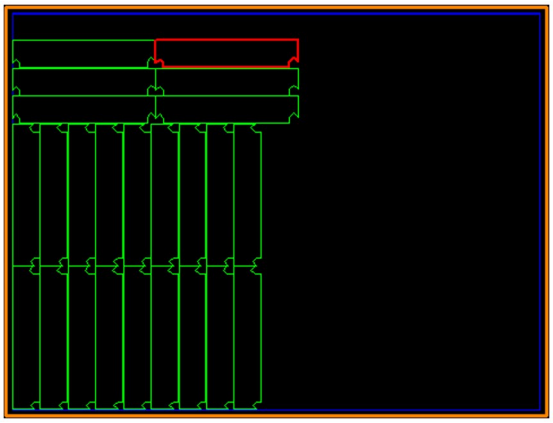

Fig. 24 Box stacking can avoid mutual hooking of box parts

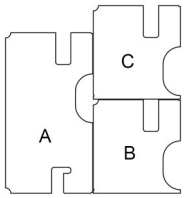

Fig. 25 Box stacking

The following details the box stacking process:

1. Box stacking of part B from the right side of part A

Move part B so that the bottom-left corner of its bounding box coincides with the bottom-right corner of part A's bounding box (Fig. 26).

Fig. 26 Stacking of part B from the right side of part A (the bottom-left corner of part B's bounding box coincides with the bottom-right corner of part A's bounding box)

If part B does not conflict with other parts or the sheet, record part B's packing attitude (positioning point and rotation angle) and centroid.

2. Box stacking of part B from the top of part A

Move part B so that the bottom-left corner of its bounding box coincides with the top-left corner of part A's bounding box (Fig. 27).

Fig. 27 Stacking of part B from the top of part A (the bottom-left corner of part B's bounding box coincides with the top-left corner of part A's bounding box)

If part B does not conflict with other parts or the sheet, record part B's attitude (positioning point and rotation angle) and centroid.

3. Box stacking of part B from the bottom of part A

Move part B so that the top-left corner of its bounding box coincides with the bottom-left corner of part A's bounding box.

If part B does not conflict with other parts or the sheet, record part B's attitude (positioning point and rotation angle) and centroid.

4. Rotate part B by 90° and repeat steps 1–3, as shown in Fig. 28.

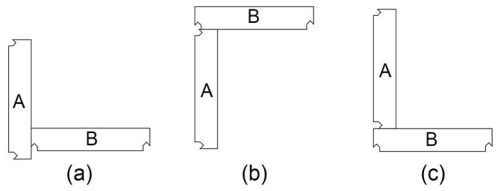

Fig. 28 Part B rotated by 90°, stacked from the right side (a), top (b), and bottom (c) of part A

5. Rotate part B by 180° and repeat steps 1–3.

6. Rotate part B by 270° and repeat steps 1–3.

7. Compare the multiple packing attitudes obtained from the above steps 1–6, and select the one with the smallest x coordinate of the centroid as the optimal attitude (assuming gravity acts horizontally to the left).

2.5 Hierarchical packing

1. Level-1 part packing

Level-1 parts refer to large rectangular parts, satisfying both of the following criteria: outer domain area $A \geq A_1$ (Table 1) and block coefficient $C_b > 0.9$.

Due to their box-like shape, level-1 parts should be packed using box stacking (as shown by the red parts in Fig. 29).

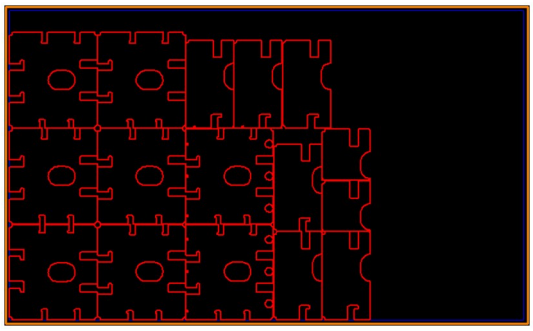

Fig. 29 Packing of level-1 parts

2. Level-2 part packing

Level-2 parts refer to large irregular parts or medium-sized parts, satisfying one of the following conditions:

(1) Large parts ($A \geq A_1$, as shown in Table 1) with a block coefficient $C_b < 0.9$;

(2) Medium parts ($A_0 \leq A < A_1$, as refer to Table 1).

As illustrated by the red parts in Fig. 30, shape matching is the primary method for level-2 parts. When satisfactory shape matching quality is not achieved (SMC<SMC_min), the following rules apply: large irregular parts are processed using box stacking, and medium parts are processed using gravity packing.

3. Level-3 part packing

Level-3 parts refer to small parts (outer domain area $A < A_0$, as refer to Table 1) and those that failed to pack in the previous two levels. Their packing strategy is similar to that of level-2 medium parts (as shown by the red parts in Fig. 31): shape matching is the first choice as well; however, if it is unsuccessful (SMC<SMC_min), gravity packing is applied.

Comparing Figs. 31 and 32, hierarchical packing has better regularity and a higher packing density than gravity packing.

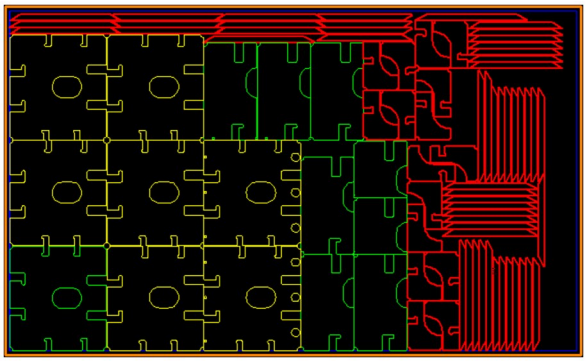

Fig. 30 Packing of level-2 parts

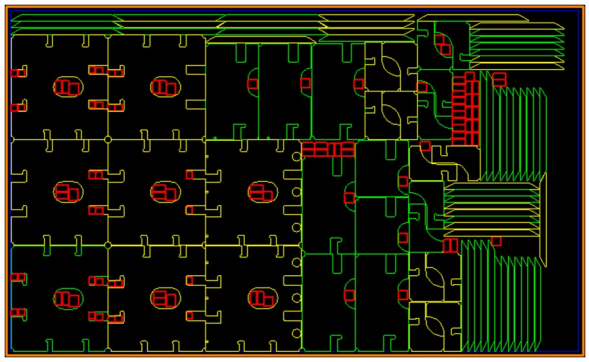

Fig. 31 Single-plate level-3 part packing, where material utilization=83.17%, the number of parts=163, and calculation time=1.52 s

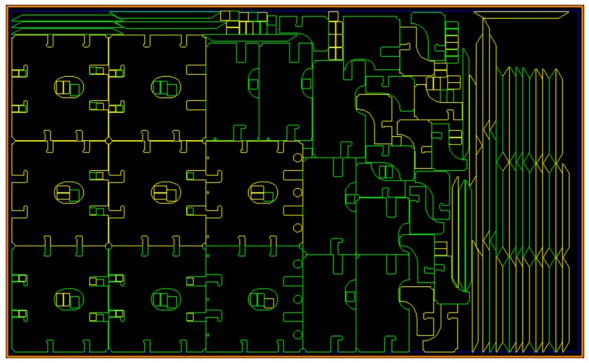

Fig. 32 Single-plate gravity packing, where material utilization=82.41%, the number of parts=163, and calculation time=10.78 s

The program in this paper was developed in Microsoft Visual Studio C++ and executed on a personal computer with a 3.7 GHz processor and 32 GB of memory. For 163 parts, the time consumed for single-plate hierarchical packing is only 1.52 s.

3 Experiments

To evaluate the hierarchical algorithm proposed in this paper, we developed a packing program and carried out two sets of experiments. The program was executed on a computer with a 3.7 GHz CPU and 32 GB of memory. The program and experiment data can be downloaded from http://www.huagongchuanhai.cn/packing/HierarchicalPacking.rar.

3.1 Experiment 1

As shown in Figs. 33 and 34, the 1400 parts used in the first set of experiments were sourced from a shipbuilding manufacturer. Although both the hierarchical packing and gravity packing methods use 29 sheets, comparing the sheet indicated by the orange frame in the upper left corner, the former occupies about 2/5 of the sheet area, while the latter uses 5/6. It is clear that the former method achieves higher material utilization and a more compact nesting result.

We used the commercial software SigmaNest to pack this set of parts. It can be observed that SigmaNest, like the hierarchical packing algorithm, used 29 sheets. However, by examining the last sheet located in the upper-left corner in Figs. 33 and 35, it is evident that the hierarchical packing achieved a higher material utilization rate.

For a more detailed analysis, enlarged views of the local packing diagrams are provided. As shown in Fig. 36a, the hierarchical packing method achieves effective shape matching among the highlighted red parts. In contrast, Fig. 37a demonstrates that such shape matching is not attained using gravity packing. Similarly, the red parts in Fig. 36b are successfully packed using box stacking, whereas the corresponding parts in Fig. 37b do not form the box stacking configuration.

In terms of computational efficiency, the hierarchical packing completes the layout of 1400 irregular parts across 29 sheets in 157.39 s, averaging 48.3 parts per sheet with a packing time of only 5.43 s per sheet. These results indicate that the method fully satisfies practical engineering requirements.

3.2 Experiment 2

To verify the adaptability of the algorithm proposed in this paper to other engineering fields, two examples from the standard public benchmarks (ESICUP: https://github.com/ESICUP) were selected for validation. The number of parts in both examples was increased to 10 times the original number. To further enhance the persuasiveness of the verification, we conducted comparative tests using the commercial software SigmaNest.

First, the benchmark case DAGLIx10 is analyzed. By comparing Figs. 38 and 39, it can be observed that the hierarchical packing used six sheets, whereas SigmaNest required 6+1/10 sheets. The material utilization rate of the proposed algorithm is slightly higher.

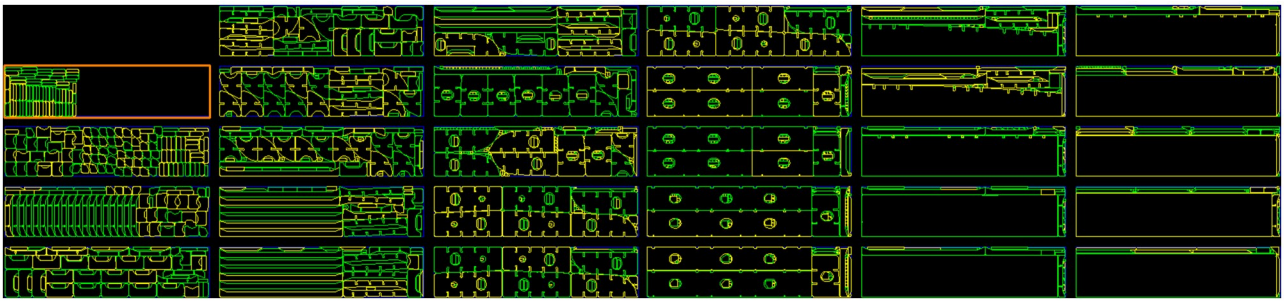

Fig. 33 Multi-plate hierarchical packing (global view), where the number of parts=1400 and computation time=157.39 s

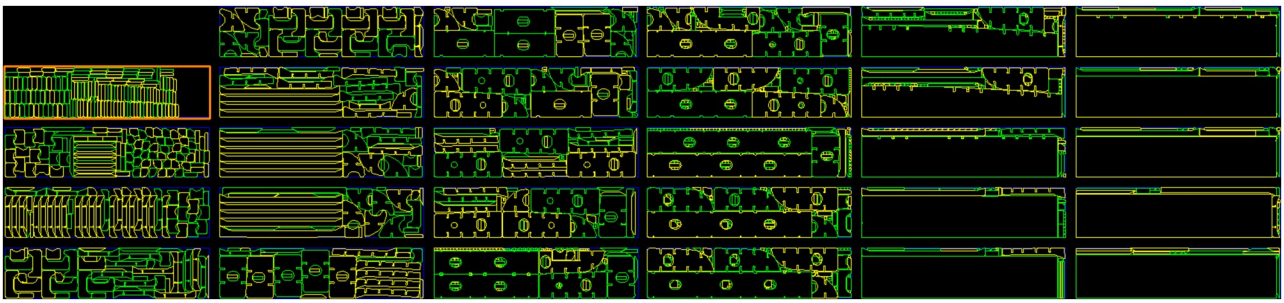

Fig. 34 Multi-plate gravity packing (global view), where the number of parts=1400 and computation time=154.31 s

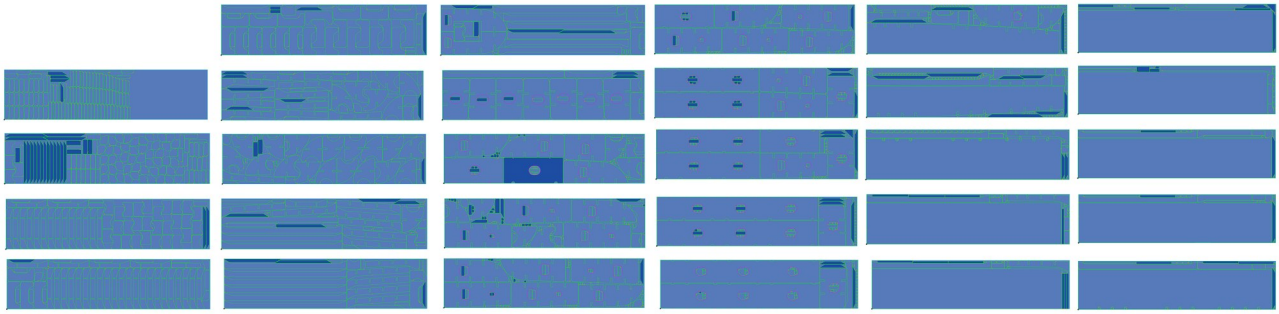

Fig. 35 Multi-plate SigmaNest packing (global view), where the number of parts=1400 and computation time=356 s

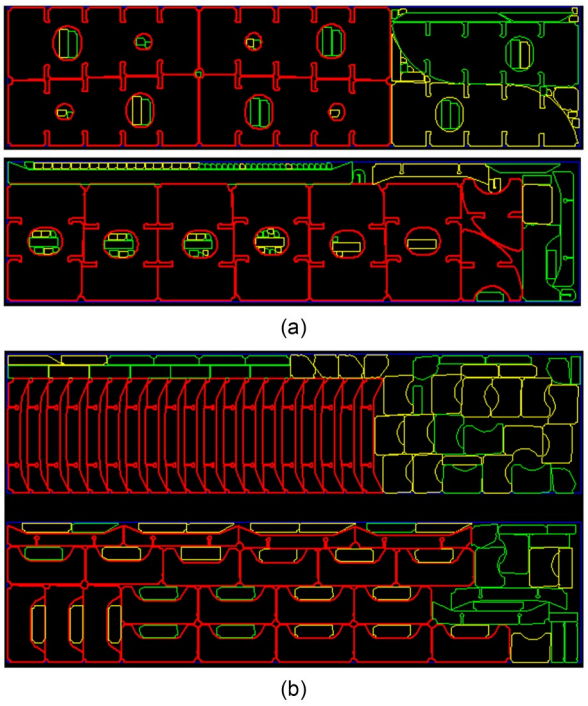

Fig. 36 Multi-plate hierarchical packing (local view): (a) shape matching; (b) box stacking

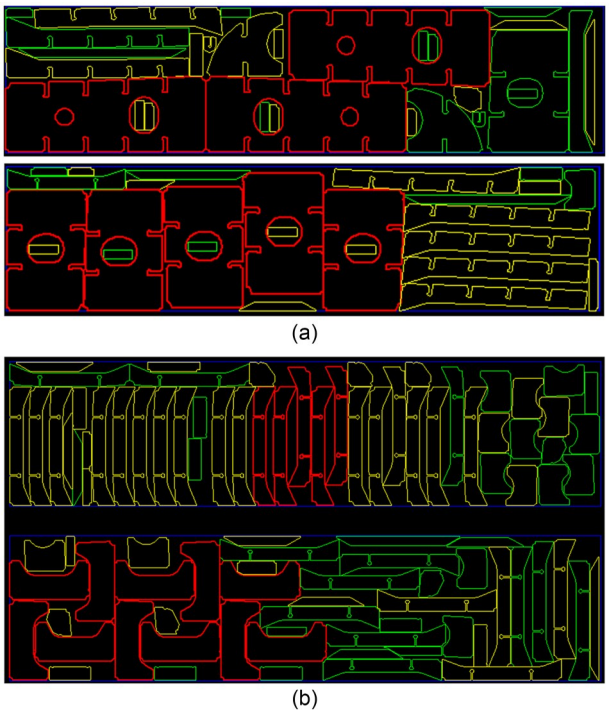

Fig. 37 Multi-plate gravity packing (local view): (a) absence of shape matching; (b) absence of box stacking

Why does the hierarchical packing algorithm achieve better results? As shown in Fig. 40a, the parts highlighted in red achieve a relatively compact arrangement through shape matching. As shown in Fig. 40b, the red parts also achieve a relatively compact arrangement using the box stacking algorithm.

For the benchmark case Marquesx10, a similar comparative analysis yields the same conclusion as noted earlier (Figs. 41–43).

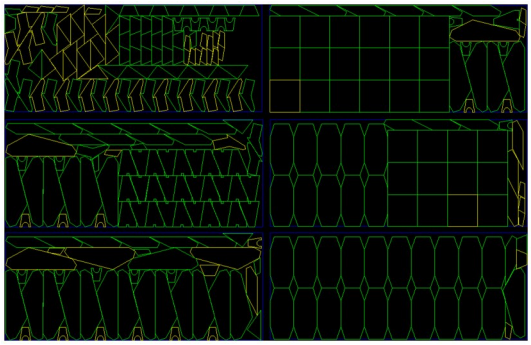

Fig. 38 Hierarchical packing (global view, DAGLIx10), where the number of parts=300 and computation time=1.23 s

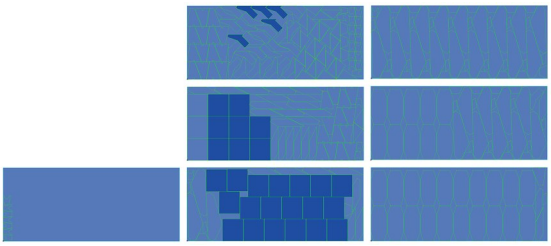

Fig. 39 SigmaNest packing (global view, DAGLIx10), where the number of parts=300 and computation time=10.40 s

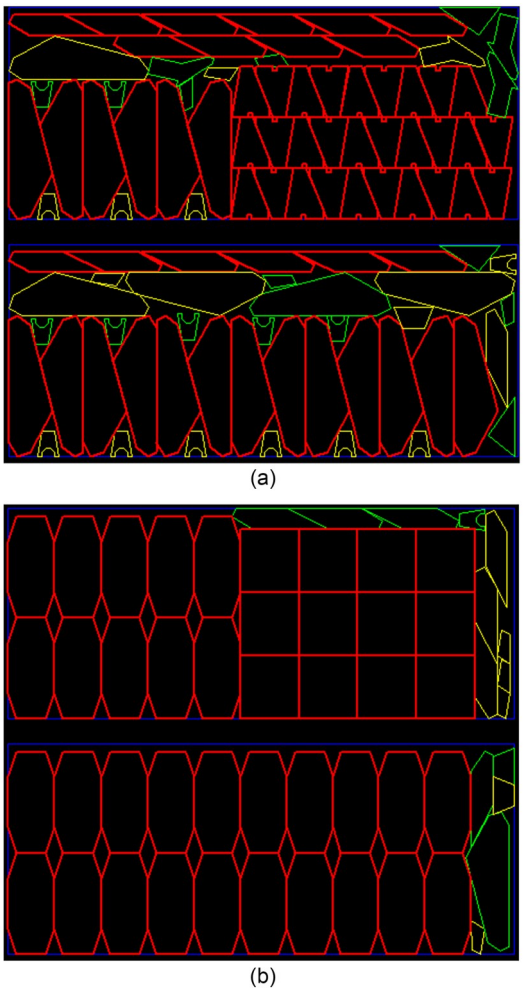

Fig. 40 Hierarchical packing (local view, DAGLIx10): (a) shape matching; (b) box stacking

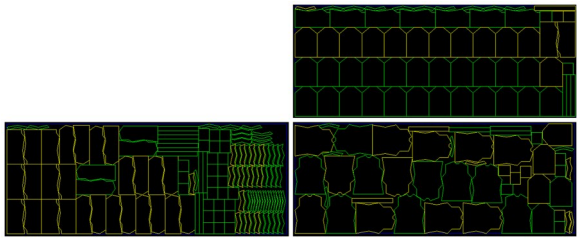

Fig. 41 Hierarchical packing (global view, Marquesx10), where the number of parts=240 and computation time=0.81 s

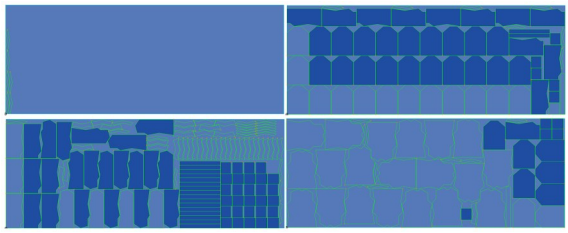

Fig. 42 SigmaNest packing (global view, Marquesx10), where the number of parts=240 and computation time=9.4 s

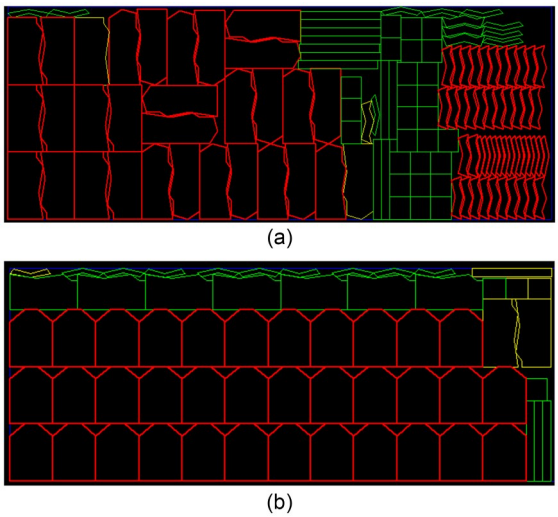

Fig. 43 Hierarchical packing (local view, Marquesx10): (a) shape matching; (b) box stacking

4 Conclusions

For decades, gravity packing has been extensively adopted for both rectangular packing and irregular packing. However, a critical limitation of this heuristic algorithm is that it prioritizes physical principles over the geometric regularity of the packing layout under certain conditions. To address this deficiency, this study develops a hierarchical algorithm that prioritizes layout regularity. Specifically, parts are first arranged to meet regularity requirements by implementing two core strategies—box stacking and shape matching. Only when layout regularity cannot be achieved by the above methods, will gravity packing be employed as an alternative.

The hierarchical algorithm not only improves material utilization by enhancing packing layout regularity but also offers an additional benefit: it significantly improves packing speed through the introduction of box stacking and shape matching algorithms. This is because box stacking is essentially rectangle packing, which is considerably faster than irregular packing. Furthermore, shape matching does not require point-by-point inspection of packing locations, which makes it much faster than the HAPE algorithm.

The hierarchical packing algorithm proposed in this paper effectively addresses large-scale packing problems, handling up to 1400 irregular parts, far exceeding the scale reported in most existing studies. With an average of 48.3 parts per sheet and an average packing time of only 5.43 s per sheet, the method fully meets the requirements for actual engineering problems.

The hierarchical packing algorithm categorizes parts into three levels by size and fullness and processes them sequentially: large rectangular parts are arranged using box stacking; large irregular parts and medium-sized parts are handled using shape matching; the remaining parts are processed either by shape matching or gravity packing. Both box stacking and shape matching contribute to improved packing regularity. Numerical examples demonstrate that pursuing packing regularity is not merely an aesthetic goal but also significantly enhances material utilization.

The shape matching algorithm employed within the hierarchical packing framework is a novel approach that simulates the manual packing process. By translating and rotating parts with similar contours, it achieves a tighter packing layout. This algorithm places a higher priority on packing regularity and material utilization, while still applying the conventional gravity packing algorithm to parts unsuitable for shape matching. Accordingly, the proposed hierarchical algorithm does not seek to overturn conventional approaches, but rather to systematically enhance them.

Acknowledgments

This work was supported by the funding from Hudong-Zhonghua Shipbuilding (Group) Co., Ltd. The author gratefully acknowledges their financial contribution and technical collaboration.

Conflict of interest

The author declares that he has no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declaration on the use of generative AI tools

During the preparation of this work, the author used DeepSeek and Doubao to improve the language. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

Albano A, Sapuppo G, 1980. Optimal allocation of two-dimensional irregular shapes using heuristic search methods. *IEEE Trans Syst Man Cybern*, 10(5):242-248. https://doi.org/10.1109/tsmc.1980.4308483

Al Theeb NA, Hayajneh MT, Jaradat MY, 2021. New strategy to improve the dotted board model for solving two dimensional cutting and packing problems. *Comput Ind Eng*, 159:107467. https://doi.org/10.1016/j.cie.2021.107467

Baker BS, Coffman EG, Rivest RL, 1980. Orthogonal packings in two dimensions. *SIAM J Comput*, 9(4):846-855. https://doi.org/10.1137/0209064

Binkley KJ, Hagiwara M, 2007. Applying self-adaptive evolutionary algorithms to two-dimensional packing problems using a four corners' heuristic. *Eur J Oper Res*, 183(3):1230-1248. https://doi.org/10.1016/j.ejor.2004.12.029

Chazelle B, 1983. The bottom-left bin-packing heuristic: an efficient implementation. *IEEE Trans Comput*, C-32(8):697-707. https://doi.org/10.1109/tc.1983.1676307

Chen M, Huang WQ, 2007. A two-level search algorithm for 2D rectangular packing problem. *Comput Ind Eng*, 53(1):123-136. https://doi.org/10.1016/j.cie.2007.04.007

Cheng SK, Rao KP, 2000. Large-scale nesting of irregular patterns using compact neighborhood algorithm. *J Mater Process Technol*, 103(1):135-140. https://doi.org/10.1016/S0924-0136(00)00402-7

Costa MT, Gomes AM, Oliveira JF, 2009. Heuristic approaches to large-scale periodic packing of irregular shapes on a rectangular sheet. *Eur J Oper Res*, 192(1):29-40. https://doi.org/10.1016/j.ejor.2007.09.012

Dowsland KA, Dowsland WB, Bennell JA, 1998. Jostling for position: local improvement for irregular cutting patterns. *J Oper Res Soc*, 49(6):647-658. https://doi.org/10.1038/sj.jors.2600563

Francescatto M, Júnior AN, 2025. Two-dimensional bin packing, cutting stock, and open dimension problems: a survey of practical requirements. *Comput Oper Res*, 183: 107209. https://doi.org/10.1016/j.cor.2025.107209

Guerriero F, Saccomanno FP, 2023. A hierarchical hyper-heuristic for the bin packing problem. *Soft Comput*, 27(18):12997-13010. https://doi.org/10.1007/s00500-022-07118-4

Guo BS, Wang YC, Wang Z, et al., 2025. Irregular multi-plate packing based on the dotted board model. *Comput Ind Eng*, 209:111439. https://doi.org/10.1016/j.cie.2025.111439

Hopper E, 2000. Two-Dimensional Packing Utilising Evolutionary Algorithms and Other Meta-Heuristic Methods. PhD Dissemination, University of Wales, Cardiff, United Kingdom.

Lai XJ, Hao JK, Yue D, et al., 2025. A heuristic algorithm with multi-scale perturbations for point arrangement and equal circle packing in a convex container. *Comput Oper Res*, 181:107099. https://doi.org/10.1016/j.cor.2025.107099

Lee WC, Ma H, Cheng BW, 2008. A heuristic for nesting problems of irregular shapes. *Comput-Aided Des*, 40(5):625-633. https://doi.org/10.1016/j.cad.2008.02.008

Liu DQ, Teng HF, 1999. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles. *Eur J Oper Res*, 112(2):413-420. https://doi.org/10.1016/S0377-2217(97)00437-2

Liu HY, He YJ, 2006. Algorithm for 2D irregular-shaped nesting problem based on the NFP algorithm and lowest-gravity-center principle. *J Zhejiang Univ-Sci A*, 7(4): 570-576. https://doi.org/10.1631/jzus.2006.A0570

Liu X, Ye JW, 2011. Heuristic algorithm based on the principle of minimum total potential energy (HAPE): a new algorithm for nesting problems. *J Zhejiang Univ-Sci A*, 12(11):860-872. https://doi.org/10.1631/jzus.A1100038

Liu YL, Zheng L, 2025. Using helical polyhedron for online irregular strip packing problem with free rotations. *Eur J Oper Res*, 327(2):407-419. https://doi.org/10.1016/j.ejor.2025.05.019

Stoyan YG, Pankratov AV, 1999. Regular packing of congruent polygons on the rectangular sheet. *Eur J Oper Res*, 113(3):653-675. https://doi.org/10.1016/S0377-2217(98)00050-2

Zhou JR, He K, Zheng JZ, et al., 2024. Geometric batch optimization for packing equal circles in a circle on large scale. *Expert Syst Appl*, 250:123952. https://doi.org/10.1016/j.eswa.2024.123952