

Research on online mixed palletizing strategy of robotic arms for multi-SKU scenarios

Guigen Jin¹, Xiaoxia Ding^{1,*}, Dian Jin²

¹ School of Logistics and Management Engineering, Yunnan University of Finance and Economics, Kunming, China

² School of Economics and Management, University of Science and Technology Beijing, Beijing, China

* strawberry110608@163.com

Abstract. In intelligent warehousing and transportation processes, the centralization of material units significantly enhances storage and handling efficiency. Among these, the centralized unitization of material pallets is in high demand and widely applied in practical operations. In multi-SKU scenarios, achieving efficient palletizing—particularly online mixed palletizing—poses a major challenge in logistics operations. This process aims to save manpower while ensuring operational efficiency. To address this issue, this paper presents a combined heuristic algorithm that integrates an anthropomorphic heuristic algorithm with a greedy algorithm incorporating local perturbations. The proposed approach accounts for constraints such as mass, volume, center of gravity, non-overlapping placement, and stability. Experimental results demonstrate that this algorithm effectively resolves the palletizing challenges for multi-SKU goods, significantly reducing space waste.

Keywords: online palletizing, multi-SKU, anthropomorphic heuristic algorithm, greedy algorithm

1. Introduction

The rise of modern e-commerce and intelligent manufacturing has driven the rapid development of the logistics industry, where palletizing plays a crucial role. In multi-SKU scenarios, achieving more efficient mixed palletizing with less manual labor and maximizing space utilization has become an urgent issue to address. The essence of palletizing lies in the three-dimensional packing problem, a type of combinatorial optimization problem [1]. As the name suggests, combinatorial optimization involves finding the optimal solution to an objective function within a defined mathematical structure and under specified constraints. Currently, one-dimensional and two-dimensional packing problems are relatively simple and thus well-studied. However, three-dimensional packing is more complex and challenging, with limited research progress. Large-scale three-dimensional packing problems cannot achieve precise optimal solutions and rely on various algorithms to obtain approximate solutions [2]. The primary methods include mathematical programming, heuristic algorithms, and hybrid approaches.

Mathematical programming is often used in small-scale packing scenarios. Common models in mathematical programming include 0-1 integer programming, mixed-integer programming, and linear programming models [3]. Fekete et al. [4] proposed a dual-tree search algorithm to solve high-dimensional packing problems, but it guarantees optimal solutions only when the number of items is fewer than 20. Similarly, Martello et al. [5] used a branch-and-bound method to solve the single knapsack problem, but this approach could only handle the optimization of up to 20 items within given time and position constraints. Heuristic algorithms are often designed based on manual packing experience. Liu et al. [6] combined tree search sub-algorithms and greedy sub-algorithms to optimize pallet loading problems. Koide et al. [7] integrated genetic algorithms, traditional Chinese truss search algorithms, and heuristic methods for packing problems and developed container loading algorithms. However, a significant limitation of these algorithms is that they only yield local optimal or approximate solutions. Hybrid algorithms, which combine two or more algorithms to leverage their combined strengths, have gained attention. Zhang Changyong et al. [8] constructed a cargo loading strategy using an anthropomorphic heuristic algorithm and then optimized the solution through crossover and mutation operations in genetic algorithms. Zhang Defu et al. [9] combined anthropomorphic heuristic algorithms with simulated annealing to optimize three-dimensional packing problems. These algorithms often use heuristic algorithms for global layout planning, followed by meta-heuristic or mathematical programming methods for local optimization [8].

In summary, using hybrid algorithms for online palletizing planning, combining anthropomorphic heuristic algorithms with greedy algorithms, can enhance global search capabilities. Therefore, this paper adopts a hybrid algorithm: an anthropomorphic

algorithm determines the loading positions, while a greedy algorithm optimizes the palletizing sequence. Additionally, a shifting operation is introduced to address gaps caused by placing small items before large ones, which may result in overhanging spaces.

2. Problem Description

This study focuses on the online mixed palletizing strategy for multi-SKU scenarios. In this context, the robotic system can only access the dimensions of certain boxes on the conveyor belt to conduct global optimal planning. Box information is sequentially identified from the conveyor belt and transmitted via camera to the online mixed palletizing algorithm, which then generates placement information. Compared to offline palletizing, online palletizing is well-suited for conveyor-line working conditions where box information is limited. It generates relatively optimal palletizing layouts within constrained timeframes, based on actual constraints.

In the logistics conveyor-line working conditions, robots handle the online palletizing of cartons for multi-SKU scenarios. The pallet is modeled as a rectangular container with a height H , length L , and width W . Essentially, the carton palletizing problem is a three-dimensional packing problem. The objective is to determine the layout that maximizes pallet space utilization while considering practical constraints for carton loading.

2.1. Constraints

In actual palletizing, numerous real-world factors introduce multiple constraints. On one hand, constraints are essential for establishing a robust optimization model. On the other hand, realistic constraints are necessary for practical engineering applications. Based on field research and analysis, the following constraints are considered:

- (1) Mass Constraint: The total mass of goods must not exceed the pallet's maximum load capacity.
- (2) Volume Constraint: The total volume of goods must not exceed the pallet's maximum loadable volume.
- (3) Center of Gravity Constraint: To ensure the stability of the palletized stack, the center of gravity must remain within a reasonable range.
- (4) Non-overlapping Constraint: No two boxes may share any overlapping points in space.
- (5) Stability Constraint: Boxes cannot be placed in a suspended manner. Each box must be supported either by other goods below it or by the base of the container. Additionally, boxes must be aligned parallel to the container, with their edges parallel to the corresponding edges of the container.

2.2. Assumptions

Due to the complexity of box stacking in actual palletizing processes, the following assumptions are made to facilitate research:

- (1) Boxes can be stacked vertically.
- (2) Boxes must be fully contained within the container.
- (3) The mass of each box is uniformly distributed, with the geometric center as the center of mass.
- (4) Boxes are not subject to deformation due to compression.
- (5) All boxes are rectangular.

3. Mathematical Model

3.1. Notation

V : Volume of the container.

n : Number of boxes.

s_i : Volume of the i -th box.

η : Pallet utilization rate.

L, W, H : Length, width, and height of the pallet, respectively.

l_i, w_i, h_i : Length, width, and height of the i -th box.

l_j, w_j, h_j : Length, width, and height of the j -th box, where $V = LWH, v_i = l_i w_i h_i$

P_i : Palletizing sequence of the i -th box.

Q_i : Order number of the i -th box.

α_i : Indicates whether the i -th box is placed on the pallet ($\alpha_i = 1$) or not ($\alpha_i = 0$).

m_i : Weight of the i -th box.

M : Maximum load capacity of the pallet.

$[G_{x_1}, G_{x_2}], [G_{y_1}, G_{y_2}], [G_{z_1}, G_{z_2}]$: Safety ranges of the center of gravity on the X, Y, and Z axes, respectively.

(g_{xi}, g_{yi}, g_{zi}) : Coordinates of the center of gravity for the i -th box.

$(x_i, y_i, z_i), (x_j, y_j, z_j)$: Coordinates of the bottom-left corner of the i -th and j -th boxes, respectively.

$\sigma_{ij}^d \in \{0,1\}$: Indicates whether boxes i and j are separated in the d -th dimension ($d=1$ for the X-axis, $d=2$ for the Y-axis, and $d=3$ for the Z-axis).

M : A sufficiently large positive number.

3.2. Box Placement Rules

3.2.1. Placement Orientation

The orientation of boxes significantly affects the final palletizing result. Assuming all boxes are rectangular, there are six possible orientations under the given constraints. Common descriptions of object orientations in daily life, such as "upright," "horizontal," "flat," or "vertical," are vague. In algorithmic applications, these six orientations must be explicitly defined. A box's orientation is determined by its alignment along three spatial dimensions: front-back, left-right, and top-bottom. A rectangular box has six faces, which can be grouped into three types based on dimensions. These three faces correspond to $A_3^3 = 6$ possible configurations, as illustrated in Figure 1.

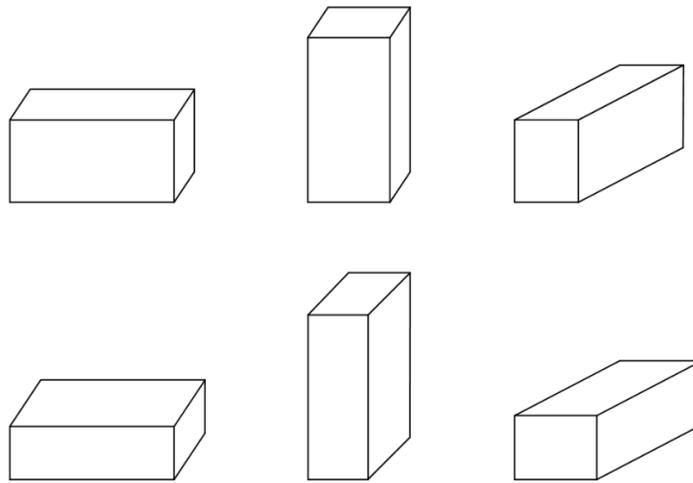


Figure 1. Diagram of Box Placement Orientations

The data structures in this study are designed based on a three-dimensional coordinate system. A fundamental data structure is a coordinate point, consisting of three ordered integers representing the projections of a point on the X, Y, and Z axes, denoted as (d_x, d_y, d_z) . Both boxes and pallets are rectangular, with their length, width, and height defined as edges parallel to the X, Y, and Z axes.

3.2.2. Container and Box Definition Rules

The container's dimensions are represented by three ordered integers: (L,W,H). As shown in Figure 2, the bottom-left corner of the container, viewed from the top-down perspective, is set as the origin (0,0,0).

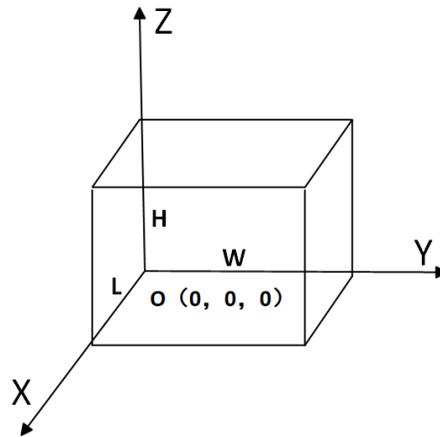


Figure 2. Diagram of Pallet Coordinates

In addition to basic positional and dimensional information, the container must store details about the boxes inside it. Box information is represented by an ordered list, which allows adding, removing, and querying operations. Since box orientations are variable, identical boxes may have different representations, as their dimensions (l,w,h) are dynamic. To ensure consistency, a box’s dimensions are expressed using an unordered set: {l,w,h}. This approach avoids storing individual length, width, and height values, instead using a dimension set and placement flag. The placement flag has a value range of {0,1,2,3,4,5}, corresponding to the six orientations.

Similarly, the position of a box in space is determined by its bottom-left corner, viewed from the top-down perspective. The position of the i-th box is denoted as (x_i, y_i, z_i) . Relevant information is illustrated in Figure 3.

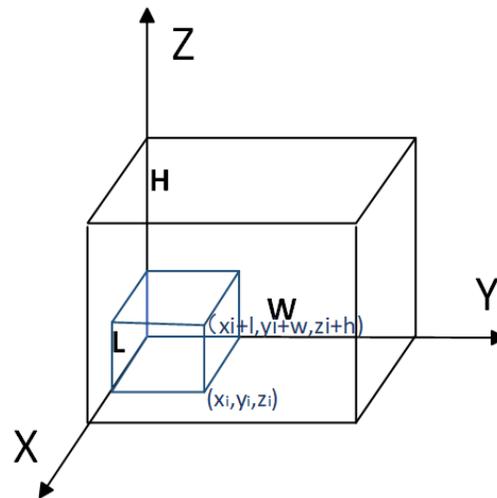


Figure 3. Diagram of Box Coordinates

3.3. Model Formulation

Based on the assumptions, constraints, and palletizing problem description, the online mixed palletizing model is formulated as follows:

Objective Function: Maximize pallet space utilization:

$$\eta = \max \frac{\sum_{i=1}^n v_i}{V} \tag{1}$$

Constraints:

$$\sum_{i=1}^n \alpha_i m_i \leq M \tag{2}$$

$$0 \leq l_i \leq L, 0 \leq w_i \leq W, 0 \leq h_i \leq H \tag{3}$$

$$0 \leq v_i \leq V \tag{4}$$

$$\begin{cases} G_{x_1} \leq \frac{\sum_{i=1}^n \alpha_i m_i g_{xi}}{\sum_{i=1}^n \alpha_i m_i} \leq G_{x_2} \\ G_{y_1} \leq \frac{\sum_{i=1}^n \alpha_i m_i g_{yi}}{\sum_{i=1}^n \alpha_i m_i} \leq G_{y_2} \\ G_{z_1} \leq \frac{\sum_{i=1}^n \alpha_i m_i g_{zi}}{\sum_{i=1}^n \alpha_i m_i} \leq G_{z_2} \end{cases} \tag{5}$$

$$\begin{cases} \sigma_{ij}^1 + \sigma_{ij}^2 + \sigma_{ij}^3 \geq 1, \forall i \neq j \\ x_i + l_i \leq x_j + M(1 - \sigma_{ij}^1) \\ x_j + l_j \leq x_i + M\sigma_{ij}^1 \\ y_i + w_i \leq y_j + M(1 - \sigma_{ij}^2) \\ y_j + w_j \leq y_i + M\sigma_{ij}^2 \\ z_i + h_i \leq z_j + M(1 - \sigma_{ij}^3) \\ z_j + h_j \leq z_i + M\sigma_{ij}^3 \end{cases} \tag{6}$$

$$\begin{cases} O_{ij} = O_x \cdot O_y > 0 \\ O_x = \max(0, \min(x_i + l_i, x_j + l_j) - \max(x_i, x_j)) \\ O_y = \max(0, \min(y_i + w_i, y_j + w_j) - \max(y_i, y_j)) \end{cases} \tag{7}$$

In the above, Equation (1) is the optimization objective to maximize pallet space utilization. Equations (2)-(5) represent the mass, volume, and center of gravity constraints. Equation (6) ensures non-overlapping placement of boxes. Equation (7) is the stability constraint, based on the overlap area between boxes.

4. Algorithm Design

4.1. Anthropomorphic Heuristic Algorithm

For addressing the multi-SKU palletizing problem, an anthropomorphic heuristic algorithm is adopted. This method involves constructing placement points, defining rules, sorting, and introducing reference planes for loading. Its advantage lies in the fact that it imposes no structural requirements on the goods. This algorithm is inspired by real-life wall-building techniques: when building a wall, a reference brick is placed first, serving as a baseline. The height of other objects is constrained by the reference brick until no more objects can be placed, at which point the height of the reference brick is adjusted.

Drawing on this concept, in the three-dimensional packing process, reference bricks are introduced in both horizontal and vertical directions to guide the loading process. This paper records placement points to identify feasible loading positions and introduces horizontal and vertical reference planes to guide the stacking process.

4.1.1. Placement Points

Placement points refer to points within the container that can serve as references for placing goods. The placement point table stores these points in an ordered list within the container's data structure. As shown in Figure 4, placement points are indicated by bold dots.

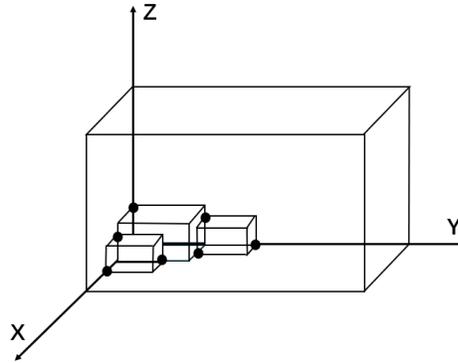


Figure 4. Diagram of Placement Points

Placement points must meet the constraints outlined earlier; if not, they cannot be used for placement. Initially, the pallet contains only one placement point, the origin. After placing box i , three new placement points are generated: $(x_i + l_i, y_i, z_i)$, $(x_i, y_i + w_i, z_i)$, $(x_i, y_i, z_i + h_i)$. The placement point table must remain ordered, which is achieved by assigning weights. The ordering is determined as follows:

- Points are sorted first by the y -coordinate in ascending order.
- If y -coordinates are equal, the x -coordinate is used in ascending order.
- If both x - and y -coordinates are equal, the z -coordinate is used in ascending order.

This sorting rule can be formalized as a weight definition or comparison logic:

$$Weight(P_k) = y_k \cdot L^2 + x_k \cdot L + z_k \quad (8)$$

Where $P_k = (x_k, y_k, z_k)$ represents the coordinates of the k -th placement point, and $Weight(P_k)$ is its weight for sorting. L is a sufficiently large positive number to represent the priority relationships between coordinates, with $L \gg \max(x, y, z)$.

4.1.2. Reference Planes

The pallet is treated as a three-dimensional container space, and two reference planes are used to decompose the palletizing problem into a single dimension. Assume the planes P_h and P_v , parallel to the XY and ZY axes, respectively, as shown in Figure 5. During palletizing, in addition to satisfying the aforementioned constraints, goods must not exceed these two reference planes. When choosing placement points, the order of selection should prioritize the y -direction, followed by the x - and z -directions. This constraint aligns with the sorting of the placement point table. Initially, both reference plane values are set to 0. If a box fails to meet the constraints during palletizing, the reference planes must be adjusted. If the constraints still cannot be met, the box cannot be placed on the pallet.

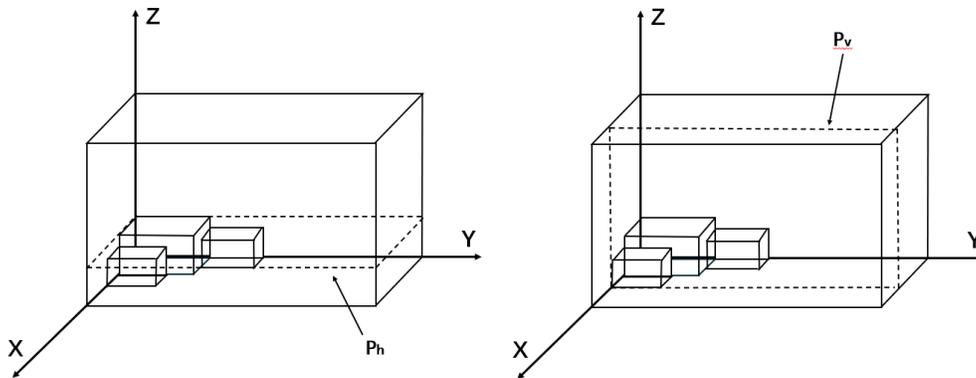


Figure 5. Diagram of Reference Planes

When using reference planes to assist palletizing, scenarios often arise where smaller boxes are placed before larger boxes, leaving gaps due to the smaller boxes not being aligned to edges. This results in suspended placements. To address this, a shifting

operation is defined: the principle is to align the boxes to edges as much as possible. Following the priority rules outlined earlier, boxes are moved first in the y-direction, followed by the x- and z-directions, provided the constraints and assumptions are satisfied.

4.2. Combined Algorithm

A combination of the anthropomorphic heuristic algorithm and a greedy algorithm is proposed. To avoid failures caused by local optima, a local random perturbation is introduced into the greedy algorithm. In this design, the anthropomorphic algorithm determines the loading positions, while the greedy algorithm decides the loading sequence.

The greedy algorithm iteratively searches for the optimal decision until convergence [10]. First, the remaining pallet space is calculated, and the loading state is initialized. The priorities of goods are then calculated. This paper assigns weights to volume, weight, and center of gravity to compute the loading sequence. The priority calculation formula is:

$$priority = 0.3 \cdot m + 0.5 \cdot v + 0.2 \cdot g \quad (9)$$

Where m is the weight, v is the volume, and g is the center of gravity of the box.

Since the greedy algorithm employs a single selection rule, it is prone to falling into local optima. To address this, a random perturbation is introduced to the priority calculation, enhancing the algorithm's adaptability. The perturbed priority formula is:

$$priority_{perturbed} = priority + \Delta P$$

$$\Delta P = \alpha \cdot random(-1,1)$$

Where ΔP is the random perturbation term, and α is the coefficient controlling the intensity of the perturbation. Random $(-1,1)$ generates a uniformly distributed random number to disrupt strict ordering.

5. Simulation Results and Visualization

The algorithm was implemented using Python, and the selected datasets were randomly generated. The datasets contained a total of 360 boxes, with the number of box specifications set at 6, 10, 12, 15, 20, and 30, respectively. The dimensions of the pallet used for testing were (1000,1200,1100) mm. The simulation results are shown in Table 1.

Table 1. Dataset Simulation Results

Test Number	Number of Specifications	Space Utilization Rate
1	6	82.5%
2	10	85.4%
3	12	87.7%
4	15	89.7%
5	20	90.2%
6	30	91.9%

The palletizing results for these six scenarios were visualized, as shown in Figure 6.

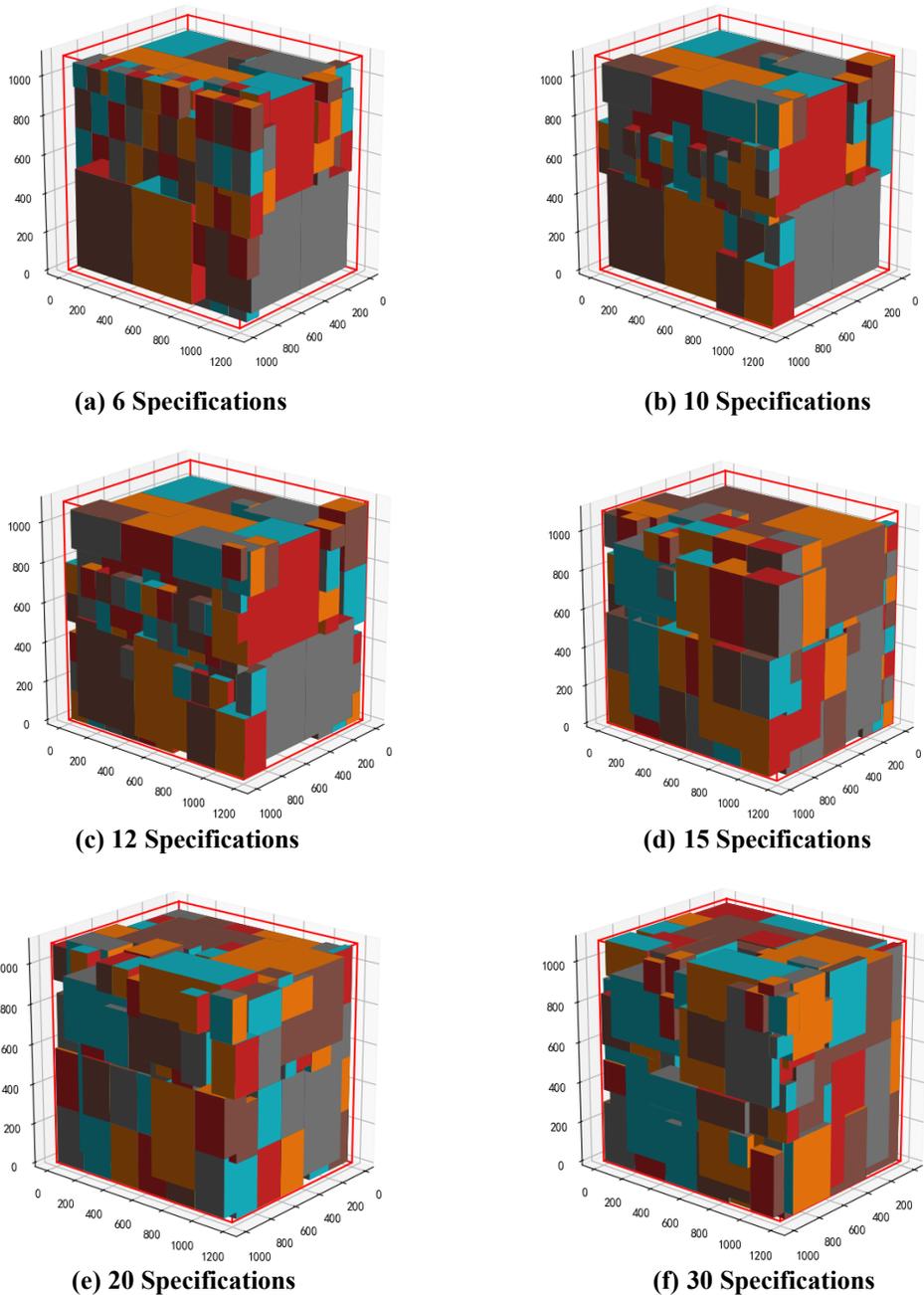


Figure 6. Visualization of Palletizing Results

6. Conclusion

This paper addresses the problem of mixed palletizing in logistics scenarios. By establishing assumptions and constraints, the palletizing problem is transformed into a three-dimensional packing problem. A combined algorithm integrating an anthropomorphic heuristic algorithm and a greedy algorithm with local perturbations was introduced. The algorithm was implemented using Python and tested on six datasets with varying specifications. The experimental results demonstrate that the combined algorithm performs well and exhibits good stability in handling scenarios with multiple specifications. As the number of specifications increases, space utilization improves. For datasets with 30 specifications, the space utilization rate reached 91.9%. This indicates that the algorithm adapts effectively to palletizing scenarios involving a high degree of heterogeneity and ensures robust palletizing performance for diverse cargo types. In practical applications, this algorithm can save space in logistics warehousing and transportation, improve space utilization, and reduce operational costs.

References

- [1] Yue, M., & Li, R. (2014). Introduction to Combinatorial Optimization [M]. Science Press.
- [2] Lin, Y. (2023). Research and Application of Online Three-Dimensional Packing Algorithms [D]. China University of Mining and Technology.
- [3] Hifi, M., Kacem, L., Negre, S., et al. (2010). A linear programming approach for the three-dimensional bin packing problem. *Electronic Notes in Discrete Mathematics*, 36, 993–1000.
- [4] Fekete, S. P., Schepers, J., & Van der Veen, J. C. (2007). An exact algorithm for high-dimensional orthogonal packing. *Operations Research*, 55, 569–587.
- [5] Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research*, 48, 256–267.
- [6] Liu, S., Zhao, H., Dong, X., et al. (2016). A heuristic algorithm for container loading of pallets with infill boxes. *European Journal of Operational Research*, 252(3), 728–736.
- [7] Koide, S., Suzuki, S., & Degawa, S. (1995). A palletize-planning system for multiple kinds of loads using GA search and traditional search. In Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human-Robot Interaction and Cooperative Robots (Vol. 3, pp. 510–515). IEEE.
- [8] Zhang, C., Zhai, Y., Zhang, Q., et al. (2021). Research on the air cargo loading problem considering loading sequence constraints. *Packaging Engineering*, 42(01), 150–156.
- [9] Zhang, D., Wei, L., Chen, Q., et al. (2007). A hybrid heuristic algorithm for the three-dimensional bin packing problem. *Journal of Software*, 9, 2083–2089.
- [10] Cai, Z., Jia, F., Li, Z., et al. (2021). Research on improving WTB communication reliability of CR200J-3 power concentrated EMU. *Locomotive and Electric Transmission*, 1, 98–103.