Packaging Model Optimization Based on Spatial Segmentation and Chaotic Firefly Algorithm

Yue Yu *

School of Ecnomics, WuhanUniversity of Technology, Wuhan, China

* Corresponding Author Email: yuelol040915@gmail.com

Abstract. In this paper, we propose a 3D box optimization model based on space partitioning algorithm and chaotic firefly algorithm, focusing on the synergistic application of the underlying layflat strategy,packing strategy and intelligent optimization algorithm in complex packaging problems. First, the problem is formalized as a 3D-MBSBPP problem by establishing an average loading rate model, defining the key mathematical conditions such as volume constraints, quantity constraints, and box type selection constraints. Secondly, design the space partitioning algorithm based on the underlying tiling strategy and packing strategy to realize high-density crating through pre-packing processing, dynamic subspace partitioning with a selection mechanism, and introduce a space merging strategy to avoid the fragmentation problem. Finally, the chaotic firefly optimization model is constructed, the Tent chaotic mapping is used to initialize the population to enhance the global search capability, and the box specifications are optimized iteratively through the veneer rate threshold constraint and the fitness function. The model realizes dynamic volume allocation through adaptive space partitioning strategy, and its mixed integer programming framework effectively balances computational efficiency and solution accuracy, which improves the average loading rate by 3.62% in the standard test cases, showing strong engineering applicability.

Keywords: Spatial segmentation algorithm; chaotic firefly algorithm; average loading rate; veneer rate.

1. Introduction

This paper focuses on the bottleneck of space utilization and computational efficiency in three-dimensional crating optimization problems, aiming to solve the problem of traditional crating algorithms that are difficult to balance loading density and solution speed under complex constraints [1]. Current mainstream crating methods mostly adopt a single heuristic strategy, which is difficult to deal with the coupling of multi-specification cargo combination optimization and dynamic space partitioning at the same time [2]. Aiming at this challenge, this paper proposes a hybrid optimization model to achieve the accurate solution of high-dimensional crating problems through the synergistic mechanism of space partitioning algorithms and intelligent optimization algorithms [3]

First, the crating planning model is constructed based on the 3D-MBSBPP problem, defining the cargo placement constraints through 0-1 integer planning and introducing the average loading rate as the core optimization objective [4]. Secondly, the space partitioning algorithm is designed to realize dynamic volume management, the pre-packing strategy is adopted to merge isomorphic goods, and the space utilization is improved through the subspace optimization mechanism and merging strategy [5]. On this basis, the chaotic firefly algorithm is constructed to optimize the packing box specifications, the Tent chaotic mapping is used to enhance the population diversity, and the size matching is ensured by the threshold constraint of the veneer rate. In particular, this paper proposes the bottom tiling strategy to enhance the stability of the boxes, and balances the computational complexity and the solution accuracy through the adaptive space partitioning mechanism. Experimental results show that the model improves the average loading rate by 3.62% in the standard test case, and its hybrid optimization framework provides an efficient solution to the large-scale crating problem [6].

2. Construction of the Model for Packing Box Planning

Regarding the design of the packing method, this is a 3D-MBSBPP problem (Three-Dimensional Mixed Bin-Sized BinPacking Problem). We need to build a packing box planning model to construct a collection of orders, boxes, and goods, and set 0-1 variables to differentiate the order usage. Then, design the poses of the goods in the box, and finally set the constraints and design the packing strategy of the goods in the box by algorithm.

2.1. Three-dimensional Spatial Crating Problem Based on Average Loading Rate

The problem studied in this paper is to design a crating scheme for a batch of cargo orders. Each order contains products with both ambient and refrigerated temperature requirements, so the appropriate ambient and refrigerated boxes need to be selected for crating. With the objective of maximizing the average loading rate of the orders, we select the appropriate box type for each order and optimize the crating scheme while satisfying various practical constraints.

To solve the above problem, the following mathematical model is developed:

The goal is to maximize the average loading rate R_j of the boxes used for order O_j, using the room temperature carton as an illustrative object, and the planning of the crating of the foam box is carried out at the same time as it:

$$R_{j1} = \frac{\sum_{i=1}^{n_{oj}} q_{ji} \times v_i}{\sum_{k=1}^{n_1} Q_{1jk} \times V_{1k}} \tag{1}$$

The model contains the following key constraints:

Volume constraints:

$$\sum^{n_{oj}} n_{1ki} \times v_i \le V_{1k}, \quad \forall j,k \tag{2}$$

This constraint ensures that the sum of the product volumes in each case does not exceed the volume of the case.

Quantitative constraints:

$$q_{ji} = \sum_{k=1}^{n_1} n_{1ki} \times Q_{1jk}, \quad \forall j,i \tag{3}$$

This constraint ensures that the quantity of each product in order O_j is equal to the sum of the quantity of that product in all the boxes used.

Box selection constraints:

$$\sum_{k=1}^{n_1} a_{jk} = 1, \quad \forall j \tag{4}$$

This constraint ensures that at most one type of carton and foam box can be selected within an order.

Logical constraints:

$$a_{jk} = \phi\left(Q_{1jk}\right) = \begin{cases} 1 & , Q_{1jk} > 0 \\ 0 & , Q_{1jk} = 0 \end{cases} \tag{5}$$

This constraint ensures that the variable in question takes the value 1 only when a box is used.

Placement constraints:

In three-dimensional space, the placement of the product pi in the box is determined by the coordinates of the lower left rear corner point $(x_{kib}, y_{kib}, z_{kib})$, and the coordinates of its upper right front corner:

$$\left(x_{kif}, y_{kif}, z_{kif}\right) = \left(x_{kib} + \Delta x_{ki}, y_{kib} + \Delta y_{ki}, z_{kib} + \Delta z_{ki}\right) \tag{6}$$

Where, $\Delta x_{ki} = l_i$, $\Delta y_{ki} = w_i$, $\Delta z_{ki} = h_i$ depends on the direction in which the product is placed.

In summary, the model maximizes the average loading rate of the boxes by optimizing the order crating scheme, under the premise of satisfying various constraints, which provides an effective theoretical support for the logistics and warehouse management of enterprises.

2.2. Spatial Segmentation Algorithm Based on Underlying Tiling and Packing Strategies

In terms of algorithmic strategies, this paper has three main focuses on crating strategies: pre-packing strategy, space partitioning optimization, and bottom tiling priority. Based on these three basic strategies, we discuss the implementation timing of the three strategies and the decision-making basis in each strategy in terms of algorithm implementation, and realize the space segmentation algorithm based on the bottom tiling strategy and packing strategy through programming.

2.2.1. Pre-packing

In the pre-processing step, in order to prevent blind space partitioning from unnecessarily decreasing space utilization, we use the pre-packing strategy - pre-packing goods of the same size or goods that are proportional in length, width and height. The advantage of such processing is that it can avoid the unnecessary increase of optimization time cost while ensuring that the narrow space is not wasted.

The installation package so generated after packing, theoretically in the installation package inside the space utilization rate reached 100%, combined with the fact that many customers buy a certain commodity quantity is generally for more than one piece. To a certain extent, this can significantly optimize the packing process.

2.2.2. Spatial segmentation and meritocracy

For an initial state of the box, there is no space occupied inside the box, and its length, width and height are labeled as L, W, H . Therefore, when loading the first goods, we prefer to place the goods in the lower left corner of the box (at the coordinates of the origin). After the first cargo is placed, the remaining space of the box immediately becomes a three-dimensional space with irregular length, width and height. For an irregular space, this algorithm strategy chooses to use the space partitioning method to divide the irregular space into multiple rectangular spaces, and completes the selection of a better lower placement subspace by choosing a space with a larger remaining subspace.

We set the length, width and height of the goods to be placed in as l, w, h. Considering the stability assumption, we have a total of two space partitioning schemes:

Segmentation based on the $Z - X$ plane. The remaining space is:

$$\text{spaceX:} \quad \{(x, y, z) |\ l \le x \le L, 0 \le y \le W, 0 \le z \le H\} \tag{7}$$

$$\text{spaceY:} \quad \{(x, y, z) | 0 \le x \le l, w \le y \le W, 0 \le z \le H\} \tag{8}$$

$$\text{spaceZ:} \quad \{(x, y, z) | 0 \le x \le l, 0 \le y \le w, h \le z \le H\} \tag{9}$$

Segmentation approach based on $Z - Y$ plane. The remaining space is:

$$\text{spaceX:} \quad \{(x, y, z) |\ l \le x \le L, 0 \le y \le w, 0 \le z \le H\} \tag{10}$$

$$\text{spaceY:} \quad \{(x, y, z) | 0 \le x \le L, w \le y \le W, 0 \le z \le H\} \tag{11}$$

$$\text{spaceZ:} \quad \{(x, y, z) | 0 \le x \le l, 0 \le y \le w, h \le z \le H\} \tag{12}$$

The two cutting methods use the $Z - X$ plane and $Z - Y$ plane as the reference plane for the first segmentation, respectively. In terms of way trade-offs, we set the partitioned subspaces to be space$X$, space Y, space Z, respectively, and their volumes are v_x, v_y, v_z. Let $V_1 = \max(v_x, v_y, v_z)$ in scheme 1 and $V_2 = \max(v_x, v_y, v_z)$ in scheme 2, if $V_1 > V_2$, it means that the largest subspace after the first partition is larger, and we choose the the first segmentation.

Similarly, after continuing to place the goods again, we use the same division and decision-making strategy to add the space with a larger volume after division to the subspace selection list.

2.2.3. Selective merging of spaces

For the above segmentation methods, if the space is divided without merging, it will continue to divide the space into pieces, which greatly affects the space utilization. Therefore, the merging of unused subspace is a very necessary inhibition in the segmentation strategy. After each loading of goods and completion of space partitioning, we make a judgment and call the space algorithm once when the following conditions are met. There are several ways to merge idle subspaces:

Merge left and right. Space a and space b can be merged into a new space o when they are adjacent to each other in the left and right positions and have the same height.

Merge front and back. Space c and space d can be merged into a new space p when they are adjacent to each other front to back and have the same height.

Merge top and bottom. When space e and space f are of the same length and width and coincide with each other, they can be merged to form a new space q.

2.2.4. Cargo placement strategy

After the pre-packing processing algorithm as well as the spatial partitioning and merging algorithms have been proposed, there are various other key points to pay attention to in the crating process. For example, in the crating process, the order of crating will directly determine whether the goods can be placed in a certain box - if the small-sized goods are loaded first, it may lead to the large-sized goods cannot be prevented; for example, how to choose the sub-space after the partition to be placed; and also for example, what are some of the strategies that can directly make the crating stability is greatly improved; all of these are the goods Placement strategies need to be discussed.

Prioritization of goods placement:

As the space of the packing box is very limited, with the placement of goods, it will lead to the internal space is divided difficult to fit any kind of goods. If placed arbitrarily, may lead to a situation: first placed smaller goods, resulting in sub-space is divided into the space does not meet the large size of the goods, so that the courier company can only make up a box, directly resulting in a significant drop in the utilization of space. Therefore, before loading the goods, we should be in accordance with its volume from large to small sequential order, if there is a volume of the same situation, we choose the largest surface as the bottom area, select the larger of the two bottom area as the higher order of goods. This decision-making strategy not only ensures that the loading of goods can improve the utilization of space, but also takes into account the stability of the factors for the subsequent laying of the bottom strategy to lay a solid foundation.

Subspace selection strategy:

In the process of subspace selection, after the space partitioning above, we choose the subspace with more remaining space as the partitioning method, but the selection of the partitioning method can only determine the specification of the subspace, but not the strategy of loading the goods into the subspace. Here, we would like to propose a subspace selection strategy, whose main scheme of decision-making is to place the next goods to be placed in three subspaces space$X$, space$Y$, space$Z$, calculate the space utilization of each subspace, and then select the one with the highest subspace utilization as the subspace to be placed in for the next goods.

Subfloor laying strategies:

In the placement process, a blanket placement with subspace utilization as the decision index may give the theoretically best result, but its stability cannot be fully guaranteed. Therefore, in the process of space partitioning, we consider the stability factor, that is, to prevent the topmost space as the largest space, not to carry out the $X - Y$ plane of the partitioning method - this is the stability consideration in space partitioning. In the process of goods placement, the bottom tiling strategy as a complementary strategy based on stability, which is proposed mainly because: space partitioning algorithms often lead to the emergence of a lot of narrow space in the use of the space is difficult to be utilized, and easy to appear in the goods piled up too high, while the space below the box is still there is not yet utilized, which may not only cause the goods on top of the box to fall down, and does not take advantage of the goal of improving space utilization of the cargo box. This may not only cause the goods above the container to fall, but also does not utilize the goal of improving the space utilization of the container.

So in an order when the number of goods is small, in order to prevent the algorithm to choose the best, but instead of stacking the goods to the top of the box in one breath, leading to the emergence of unreasonable programs, we use the bottom tiling strategy to pave the bottom of the box as the optimal program, to prevent the goods in the box of a book stacked too high.

2.2.5. Frozen order processing

Since the title requires two ice cubes to be added for frozen and refrigerated orders, based on practical considerations we consider that frozen orders need to add the constraint of two ice cubes per case. In terms of algorithm implementation, if we follow the above prioritization of goods placement, it may lead to insufficient priority of ice cubes to ensure the constraint that there are two ice cubes in each case; and if we put the priority of ice cubes in the first place, we may lose the desired result.

Therefore, we believe that the location of ice cubes does not need to satisfy our assumption of goods location, and the focus should be on whether there is enough space for ice cubes after putting in the goods, so we no longer study the specific placement of ice cubes for the purpose of simplifying the algorithm, but only focus on whether the remaining space can satisfy the constraint of two ice cubes in each box. The conditions of whether the ice cubes are placed diagonally or not, and whether there is a complete support surface or not are not considered, so that we can follow the algorithm above to achieve a uniform effect.

3. Construction of the Model for Optimization of Packing Box Dimensions

3.1. Introduction and Objectives of the Model

We have defined a key metric, Fit Ratio, to measure the degree of fit between the product and the currently selected case space. Fit R_{ie} reflects how well a product fits the surface of a box in three dimensions. By calculating the dimensional deviation of the product in the x, y, and z directions, we can determine how closely the product fits the interior space of the box in each direction.
Specifically, the formula for calculating the veneer rate is as follows:

$$R_{ie} = \left(1 + r_{iez}\right)\frac{\Delta x_{ki}\Delta y_{ki}}{\Delta X_{ke}\Delta Y_{ke}} + \left(1 + r_{iey}\right)\frac{\Delta x_{ki}\Delta z_{ki}}{\Delta X_{ke}\Delta Z_{ke}} + \left(1 + r_{iex}\right)\frac{\Delta y_{ki}\Delta z_{ki}}{\Delta Y_{ke}\Delta Z_{ke}} \tag{13}$$

Where, $\Delta x_{ki}, \Delta y_{ki}, \Delta z_{ki}$ denote the dimensions of the product in the x y, z directions, respectively, and $\Delta X_{ke}, \Delta Y_{ke}, \Delta Z_{ke}$ denote the internal dimensions of the case in the corresponding directions, respectively. The lamination rates $r_{iex}, r_{iey}, r_{iez}$ are defined by the following equation.

$$r_{iex} = \begin{cases} 0, & \beta_x > 0 \\ 1, & \beta_x = 0 \\ -\infty, & \beta_x < 0 \end{cases}, \quad r_{iey} = \begin{cases} 0, & \beta_y > 0 \\ 1, & \beta_y = 0 \\ -\infty, & \beta_y < 0 \end{cases} \quad r_{iez} = \begin{cases} 0, & \beta_z > 0 \\ 1, & \beta_z = 0 \\ -\infty, & \beta_z < 0 \end{cases} \tag{14}$$

Where, $\beta_x = \Delta X_{ke} - \Delta x_{ki}, \beta_y = \Delta Y_{ke} - \Delta y_{ki}, \beta_z = \Delta Z_{ke} - \Delta z_{ki}$ denote the difference between the dimensions of the product in the x, y, and z directions and the dimensions of the interior space of the box, respectively.

3.2. Optimization Objectives and Constraints

To achieve higher loading efficiency, we set two optimization goals:
Maximize the average veneer rate R_e, i.e:

$$\max R_e = \frac{1}{n_k}\sum_{i=1}^{n_k} R_{ie} \tag{15}$$

Where, n_k is the total number of products loaded in the box?
Maximize the average loading rate R_j, i.e:

$$\max R_j = \frac{\sum_{i=1}^{n_{oj}} q_{ji} \cdot v_i}{\sum_{k=1}^{n_1} Q_{1jk} \cdot V_{1k}} \tag{16}$$

Where, q_{ji} is the quantity of product p_i in order o_j, v_i is the volume of the product, Q_{1jk} is the quantity of box c_{1k} used, and V_{1k} is the total volume of box c_{1k}.

The optimization process of the model is limited by the following constraints:

$$\sum_{i=1}^{n_{oj}} n_{1ki} \cdot v_i \le V_{1k} \tag{17}$$

$$q_{ji} = \sum_{k=1}^{n_1} n_{1ki} \cdot Q_{1jk} \tag{18}$$

$$\sum_{k=1}^{n_1} a_{jk} = 1 \tag{19}$$

3.3. Firefly Algorithm Based on Chaotic Operators

When optimizing the box specifications, in order to solve the better box adjustment scheme, we make the length, width and height of the box float by twenty percent, and use the chaotic coupling firefly algorithm to optimize the average loading rate as the goal in that range, and require the lamination rate to be higher than a certain threshold to find the better box specifications.

3.3.1. Firefly algorithm

FireflyAlgorithm (FireflyAlgorithm) is a stochastic optimization algorithm constructed by simulating the luminescence behavior of fireflies in nature, the algorithm abandons some biological significance of firefly luminescence, and only uses its luminescence characteristics to search for a partner according to its search area and move to the firefly with a better position, thus realizing the position evolution. For the FA algorithm, the following three assumptions are usually made:

Fireflies do not distinguish between males and females, and fireflies that glow stronger can attract other fireflies that glow weaker.

The attractiveness β of each firefly is proportional to its luminous intensity I; for any two luminous fireflies, the weaker luminous firefly moves toward the stronger luminous firefly, and the attractiveness and luminous intensity become smaller as the distance between the two fireflies becomes larger; the brightest firefly (i.e., the one with the largest value of I) is flying randomly The brightest fireflies (i.e., those with the largest I values) are flying randomly.

The luminous intensity I of the fireflies is related to the value of the objective function.

In terms of algorithm implementation, a few basic steps are briefly described below:

Initialize the firefly position; determine the distance between firefly i and j; update the attractiveness of firefly i; according to the attractiveness, update the position of firefly i; make the brightest firefly fly randomly; stop the computation when reaching the set threshold or the number of iterations; and vice versa to make $t = t + 1$, and return to the step 2 to continue the iteration.

3.3.2. Chaos mapping strategy (math.)

In the standard firefly algorithm, the initial positions of randomly generated fireflies are unevenly distributed, which often leads to the problems of low algorithmic optimization efficiency, slow convergence speed and large computational time cost. To solve this problem, we introduce a chaotic mapping strategy that utilizes chaotic operators with the characteristics of unrepeated traversal, randomness, and regularity to keep the algorithm with population diversity, which is used to improve the global search performance of the algorithm. Since there are many kinds of chaotic mappings, and the chaotic operators that are not used have a great influence on the algorithm's solution efficiency, Prof. Qin Honglei's literature points out that the Tent mapping has a better solution than the commonly used Logistic mapping, so we use the Tent mapping to initialize the population. The iterative formula of the Tent mapping is shown below:

$$\text{Tent}_{t+1} = \begin{cases} \text{Tent}_t / \rho & \text{Tent}_t \in (0, \rho] \\ \rho(1 - \text{Tent}_t) & \text{Tent}_t \in (\rho, 1) \end{cases} \tag{20}$$

Where, $Tent_t$ is the current value of the t th generation chaotic sequence; ρ is a control parameter, taken as $\rho = 0.5$; and the initial value $Tent_0$ is set as a random number in the interval (0, 1). After traversing the complete two-dimensional space, the chaotic particles will be mapped into the search space of the average loading rate model to obtain the improved specifications of the first generation of packing boxes, and the mapping formula is shown below:

$$P_{pos}^t = \begin{cases} x_{\min} + (x_{\max} - x_{\min}) \cdot \text{Tent}_t \\ y_{\min} + (y_{\max} - y_{\min}) \cdot \text{Tent}_t \\ z_{\min} + (z_{\max} - z_{\min}) \cdot \text{Tent}_t \end{cases} \tag{21}$$

Where, $x_{\min}$ and $x_{\max}$ are the lower bounds of $0.8L_k$ and $1.2L_k$ of the floating range of the packing box length, respectively, and y and z correspond to the upper and lower bounds of the width and height floats, in that order. P_{pos}^t Is the position of the t th generation of fireflies.

3.4. Optimization Results

According to the chaotic firefly algorithm, we set the termination condition of the algorithm as the optimization objective convergence or iteration 100 generations, and finally got the optimized packing box, the average loading rate is optimized from 0.332 to 0.344, and the percentage of improvement is 3.62%. The fitness iteration curve is shown in Fig. 1:

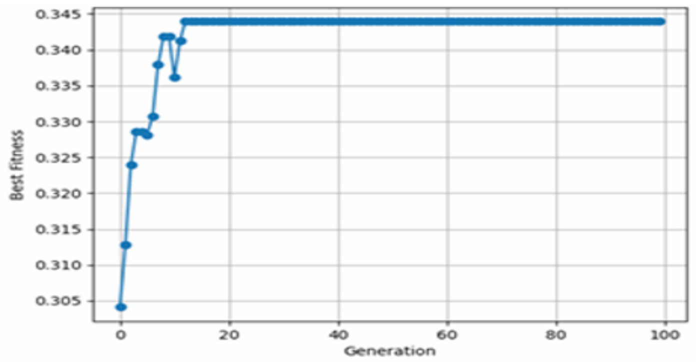

Fig 1. Iterative graph of fitness.

As shown in the figure, the target loading rate converges at generation 12, after which the optimized loading rate converges to 0.344.

The experiments showed that the most frequently used foam boxes were No. 1 and No. 2 foam boxes; the most frequently used cardboard boxes were No. 1 and No. 3 cardboard boxes. Their improved dimensions are shown in Table 1:

Table 1. Packing box specification table.

Case type	Length (cm)	Width (cm)	Height (cm)
1# Foam box (before improvements)	31.00	19.40	15.00
1# Foam box (improved)	38.00	25.00	5.50
2# Foam box (before improvements)	34.50	24.50	18.50
2# Foam box (improved)	31.00	23.50	13.50
1# Carton (before improvements)	22.00	15.00	12.00
1# Carton (improved)	27.50	19.00	4.50
3# Carton (before improvement)	30.00	24.00	15.50
3# Carton (improved)	33.50	28.00	7.50

4. Conclusion

The spatial partitioning algorithm based on the underlying tiling strategy and packing strategy proposed in this paper, combined with the chaotic firefly optimization method, achieves remarkable results in the 3D crate optimization problem. The experimental results show that the algorithm achieves a 3.62% increase in average loading rate through pre-packing processing, dynamic space partitioning and optimal merging strategy, which verifies its efficiency under complex spatial constraints. First, the average loading rate model provides a quantitative evaluation framework for the problem through the mathematical expression of volume constraints, quantity constraints and box selection constraints. Second, the space partitioning algorithm achieves dynamic optimization of space utilization through the Z-X/Y plane partitioning strategy and the subspace utilization maximization criterion, and its volume calculation function and merging logic ensure the stability of the algorithm. Finally, the chaotic firefly algorithm optimized the parameters of packing box specifications through Tent mapping initialization population and attractiveness updating mechanism, and its fitness curve converged to a loading rate of 0.344, which proved the effectiveness of the algorithm in parameter search. Future research can further explore the integration of multi-objective optimization and reinforcement learning to cope with higher dimensional packing constraints and dynamic order scenarios.

References

[1] Liu Hua, Wang Wei, Zhang Jie. Research on optimization strategy of packing box selection and space utilization [J]. Logistics Technology, 2022, 41(3). 45-50.

[2] Chen Wei, Wang Yong. Research on optimization of logistics box selection based on genetic algorithm [J]. Computer Integrated Manufacturing Systems, 2021, 27(2): 125-131.

[3] Zhang Lei, Wang Li. Optimization method of packing box selection and space utilization in e-commerce logistics [J]. Logistics Engineering and Management. 2020, 42(5): 112-118.

[4] Wang Qiang, Li Na. Mathematical modeling and optimization study of the crate loading problem [J]. Operations Research and Management, 2019, 28(4): 96-103.

[5] Zhao Peng, Liu Jie. Research on e-commerce packing box selection based on multi-objective optimization [J]. Logistics Technology, 2023, 44(1): 89-95.

[6] Wang, B., & Dong, H. Bin Packing Optimization via Deep Reinforcement Learning [J]. arXiv, 2024, 2403.12420.