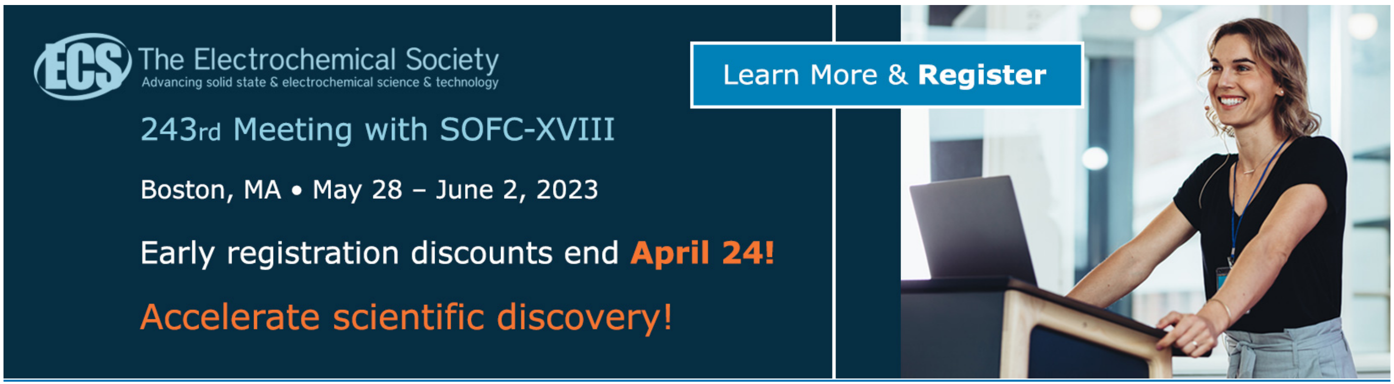
A Hybrid Algorithm for Two-dimensional and Three-stage Exact Packing Optimization Problem


The Electrochemical Society
Advancing solid state & electrochemical science & technology
Learn More & Register
243rd Meeting with SOFC-XVIII
Boston, MA • May 28 – June 2, 2023
Early registration discounts end April 24!
Accelerate scientific discovery!

A Hybrid Algorithm for Two-dimensional and Three-stage Exact Packing Optimization Problem

Yifan Sun, Chuang Zhang, Rui Liu, Limin JIA, Yong QIN, Zhipeng WANG*

State Key Lab of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China

zpwang@bjtu.edu.cn

Abstract. In this paper, we study the packing optimization problem of square plates under the conditions that the cutting mode is guillotine cut, the number of cutting stages cannot exceed 3, and the order sequence is not considered. Because it is difficult to determine the constraint conditions in the process of stacking items into stacks, we divide the nonlinear integer programming model into two stages to establish: one stage is the process of stacking items into stacks followed by splicing them into stripes, and the other stage is the process of forming stripes. Finally, a packing optimization algorithm is proposed to solve the problem, which combines an improved bottom-up left-justified algorithm (BL algorithm), greedy algorithm, and iterative sequential value correction algorithm (ISVC algorithm) with a genetic algorithm (GA) as the core. Using this algorithm to solve the data after data preprocessing, the average plate utilization rate can reach 89.29%, which is 10.42% more than the final data without any processing.

1. Introduction

The packing optimization in the personalized customization production mode is essential. The purpose of optimization is to reasonably plan the layout of square parts on the plate so as to reduce the plate waste problem in the blanking process. In this paper, we mainly discuss the packing optimization problem under the following conditions: the number of cutting stages does not exceed 3, that is, items are stacked into stacks, stacks are spliced into stripes, and finally, stripes form plates; the layout scheme is 3H (the length and width are the same) and 3E (one side is equal in length and width) in guillotine cut; the specification of the square plate is $2{,}440 \times 1{,}220\text{mm}^2$, and the quantity is sufficient. And the influence of saw slot width is not considered. The data is from dataset A of question b of *the 19th China Postgraduate Mathematical Modeling Competition "China Optics Valley · Huawei Cup."*

First, we establish a nonlinear integer programming model in two stages. To solve the model, we propose a hybrid algorithm for packing optimization. Suppose the item is directly used as a chromosome, and GA[1~2] is used to perform the entire array operation. The dimension of the action object will be too large, which will affect the time complexity of the solution process and the effective utilization of the plate. So we use different algorithms to arrange items in each process: in the process of stacking items into stacks, we use the improved BL algorithm[3] and stack items in 3H and 3E successively; later, the ISVC algorithm[4] is used to place the stacks in the process of splicing the stacks into stripes; at the same time, the items that cannot be stacked in 3H and 3E mode are directly formed into stacks according to greedy algorithm[5~6]. After the above operations, the chromosomes used to construct plates with GA have changed from item to stack. It dramatically reduces the dimension of the entire arrangement. Not only can the whole arrangement be carried out quickly, but it also can obtain a locally optimal solution that is closer to the global optimal solution.

In actual processing, we found that the order of the data will also affect the effective utilization of the algorithm, so we first preprocessed the original data by reorganizing and sorting, compared the results of the processed data with those of the non-processed data at the end, and determined that the preprocessing in descending order according to the side length will make the effective utilization of the algorithm higher.

2. Data preprocessing

GA is closely related to the data order of chromosomes (chromosomes here refer to item), so the order of data significantly affects the effective utilization of plates. Therefore, we propose two ideas to rearrange the data order:

- Instead of processing the original data set, we directly use the algorithm to solve the problem.
- First, item_length and item_width are exchanged and stitched with the original data. Then, classify them into groups according to the same item_length in the dataset. Then, remove the items with the same ID in the same data type and preferentially remove the same ID between different groups from the group with a small number of items.

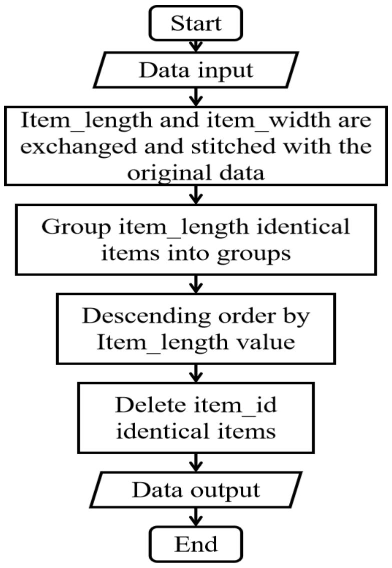

Start
Data input
Item_length and item_width are exchanged and stitched with the original data
Group item_length identical items into groups
Descending order by Item_length value
Delete item_id identical items
Data output
End

Figure 1. Flow chart of data preprocessing.

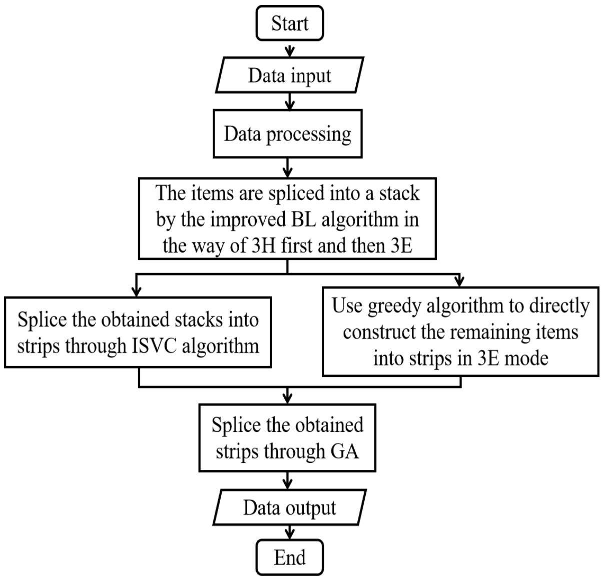

Start
Data input
Data processing
The items are spliced into a stack by the improved BL algorithm in the way of 3H first and then 3E
Splice the obtained stacks into strips through ISVC algorithm
Use greedy algorithm to directly construct the remaining items into strips in 3E mode
Splice the obtained strips through GA
Data output
End

Figure 2. Flow chart of packing optimization algorithm.

Then, we choose two methods to sort groups: one is to sort groups in descending order according to the item_length of different groups, and the other is to sort groups according to the number of items in each group. Through a large number of experiments, we found that the data processed according to the first sorting method had a better effect. Therefore, the grouped data should be sorted in descending order of item_length. The flow chart of data preprocessing is shown in Figure 1.

3. Establishment of the nonlinear integer programming model

The model of the packing optimization problem is actually a nonlinear integer programming model. We constrain the stack and stripe through two stages, respectively.

First, establish the model in the first stage, and introduce two 0-1 decision variables x_{ij} and y_i, where

$$x_{ij}=\begin{cases}1, & \text{the } j\text{-th item in group } i \text{ is stacked in the stack,}\\ 0, & \text{the } j\text{-th item in group } i \text{ is not stacked in the stack.}\end{cases}$$

$$y_i=\begin{cases}1, & \text{the } i\text{-th stack is spliced into the stripe,}\\ 0, & \text{the } i\text{-th stack is not spliced into the stripe.}\end{cases}$$

Suppose that a_{ij} represents the width of the *j*-th item in group *i*, b_i represents the length of group *i*, and the number of groups and items in each group are K and M_i, respectively, so $i\in\{1,2,\ldots,K\}$, $j\in\{1,2,\ldots,M_i\}$.

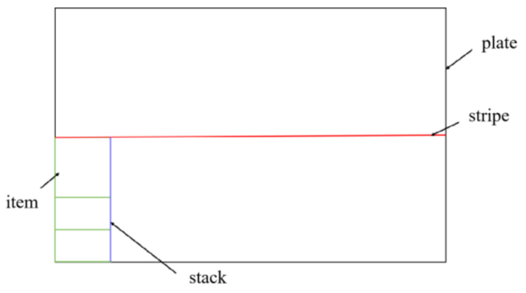


Figure 3. Schematic diagram of stacking items into stacks.

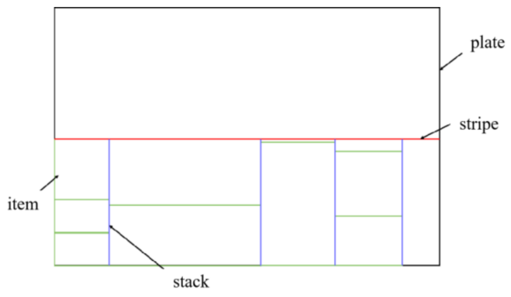


Figure 4. Schematic diagram of stacks spliced into stripes.

The first stage model has two main constraints:

The same item_length in a group should be close to the 2,440mm boundary of the plate, while the item_width should be close to the 1,220mm boundary. The schematic diagram is shown in Figure 3. This completes the operation of stacking stacks from items. The only constraint for stacking stacks is the edge length of 1,220mm, so the first constraint can be expressed as $\sum_{j=1}^{M_i} a_{ij}x_{ij}\le 1220, i=1,2,\ldots,K$.

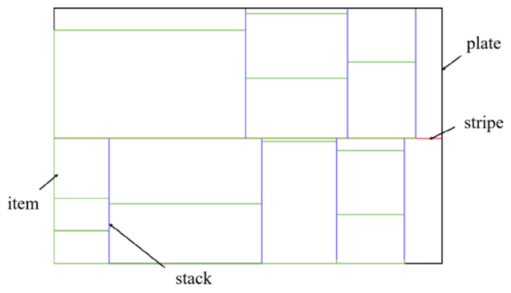
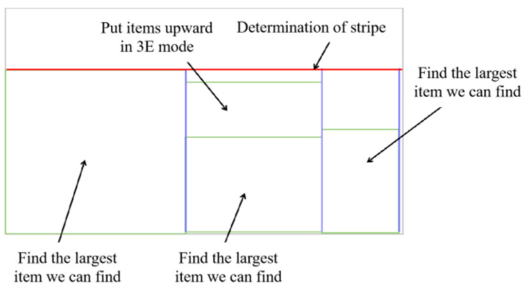


Figure 5. Schematic diagram of stripes forming plates.


Put items upward in 3E mode
Determination of stripe
Find the largest item we can find
Find the largest item we can find
Find the largest item we can find
Figure 6. Schematic diagram of splicing the remaining items into stripes.

Then splice the stack from left to right on the boundary of 2,440mm of the plate. The schematic diagram is shown in Figure 4. In this way, the operation of splicing stripes from the stack is completed. Therefore, the constraint condition at this time is to splice as many stacks as possible in the same stripe without exceeding the edge length of the plate. Then the second constraint condition can be expressed as $\sum_{i=1}^{K} b_i y_i \le 2440$.

The objective function of the first stage is to maximize the utilization area of each stripe, that is $\max\sum_{i=1}^{K} b_i y_i \left(\sum_{j=1}^{M_i} a_{ij} x_{ij} \right)$.

To sum up, the nonlinear integer programming model in the first stage is:

$$\begin{aligned} &\max\sum_{i=1}^{K} b_i y_i \left(\sum_{j=1}^{M_i} a_{ij} x_{ij} \right) \\ \text{s.t. } &\sum_{i=1}^{K} b_i y_i \le 2440, \\ &\sum_{j=1}^{M_i} a_{ij} x_{ij} \le 1220, j = 1,2,\cdots,K, \\ &x_{ij} \in \{0,1\}, \\ &y_i \in \{0,1\}. \end{aligned}$$

Next, we will build the model in the second stage, that is, the process of splicing stripes into plates. At this stage, only one 0-1 decision variable z_k needs to be introduced, which means:

$$z_k = \begin{cases} 1, & \text{the } k\text{-th stripe is spliced into the plate,} \\ 0, & \text{the } k\text{-th stripe is not spliced into the plate.} \end{cases}$$

Suppose c_k represents the length of the short side of the *k*-th stripe, and the number of stripes generated in the first stage is N, so $k \in \{1,2,...,N\}$.

The process of splicing stripes into the plate can only be limited by the 1220mm side length, as shown in Figure 5. The constraint condition can be expressed as $\sum_{k=1}^{N} c_k z_k \le 1220$.

The objective function of the second stage is to maximize the sum of the short edge lengths of all stripes that make up a plate, that is $\max\sum_{k=1}^{N} c_k z_k$.

Then the nonlinear integer programming model of the second stage is:

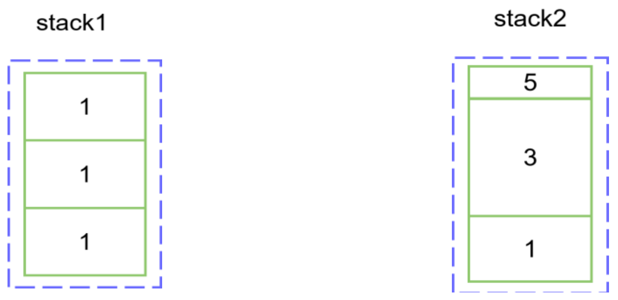
$$\begin{aligned} &\max\sum_{k=1}^{N} c_k z_k \\ \text{s.t. } &\sum_{k=1}^{N} c_k z_k \le 1220, \\ &z_k \in \{0,1\}. \end{aligned}$$

4. Packing optimization algorithm

Taking Data A1 as an example, if we want to maximize the utilization rate of plates, we need to traverse all 752 data. Therefore, this nonlinear integer programming model is not only computationally expensive and time-consuming but also difficult to find the optimal value.

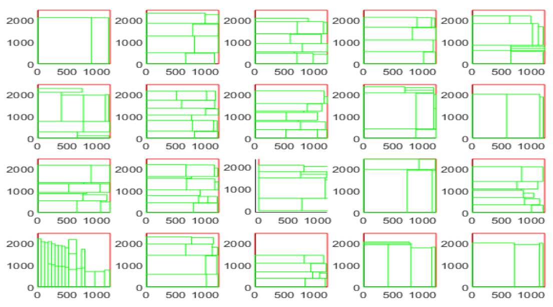
Therefore, in order to improve the efficiency of the algorithm, we propose a packing optimization algorithm based on GA combined with an improved BL algorithm, greedy algorithm, and ISVC algorithm, following the principle of the minimum number of plates used. The flow chart of this algorithm is shown in Figure 2.

First, preprocess the data and specify the item_ length is length, and item_ width is width.



(a)3H mode. (b)3E mode.
Figure 7. Schematic diagram of two exact layout modes.

Then, to ensure the cutting method of guillotine cut, we propose an improved BL algorithm to stack items into stacks in the way of 3H first and then 3E. The schematic diagram of 3H and 3E modes is shown in Figure 7, where the same number represents that the length and width of two items are equal. The specific idea of the improved BL algorithm is: from the bottom left corner, stack from bottom to top until they exceed the boundary, which constitutes a stack. Whether 3H or 3E, one side of two bottom-up items must have the same length, ensuring that each stack meets the requirements of a guillotine cut.



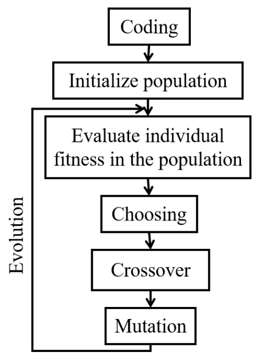


Figure 8. Partial result plots for packing optimization. Figure 9. flow chart of GA.

In the process of splicing stacks into stripes, we use the ISVC algorithm. This algorithm is not only beneficial to the stack layout of the current stripe but also retains a certain number of stacks with small areas that are easy to lay out. Then, we can find the packing scheme with the best utilization rate. And the general idea of the ISVC algorithm is: because the stacks with large differences in length will lead to waste when forming a stripe, we form the stripe in the order of stack length from large to small until it exceeds the length of the plate

At the same time, the greedy algorithm is used to directly splice the remaining items into stripes according to the 3E method. The principle of the greedy algorithm is to always consider the item with the largest area first, not the overall optimization, but only the local optimization. The specific splicing method of this process is as follows: first, find the item with the largest area, then treat it as a stack, and then place it on the left of the stripe. We specify that the width of the largest area is the width of the stripe and then place the remaining item with the largest area in the stripe. If they cannot be placed, the stripe will be output. In this process, follow the 3E method to stack the next item into a stack, that is, search for items with equal side lengths that can be placed on the top. If yes, the two items will be stacked into a quasi-stack. If not, the item will be in a separate stack. Then the right side is spliced until the area on the right side is not enough to put down any item so that the operation from item to stripe can be completed. The schematic diagram of this process is shown in Figure 6.

Finally, the stripe is spliced into a plate. After the above two steps, the number of stripe dimensions constituting the plate is far less than 752. Then, the GA is used to fully arrange them to get the preliminary output results, which significantly improves the efficiency of the algorithm and makes it easier to find the optimal solution. We first encode the data, randomly arrange the generated stripe data and express them as chromosomes, and then evaluate the individual fitness, selection, crossover, mutation, and evolution of the stripe population according to the algorithm flow chart in Figure 10. In this paper, the number of evolutions is set to 500.

We use the packing optimization algorithm to solve the preprocessed data and compare the results with those of the data without preprocessing. See Table 1 for the comparison results. And it shows that the packing optimization algorithm is better for grouped and descending data. Therefore, the data should be preprocessed accordingly. Some preliminary renderings of the packing optimization algorithm are shown in Figure 8.

Table 1. Comparison of preprocessing and non-preprocessing results.

	Results without preprocessing		Results with preprocessing	
	Number of plates	Plate utilization rate	Number of plates	Plate utilization rate
Data A1	106	78.81%	95	87.94%
Data A2	107	77.45%	95	87.24%
Data A3	105	79.74%	92	91.01%
Data A4	103	79.47%	90	90.95%

5. Conclusion

This paper describes a packing optimization algorithm based on GA and combined with an improved BL algorithm, ISVC algorithm, and the greedy algorithm. The algorithm can be used to optimize the layout problem under the condition that the guillotine cut, the number of cutting stages does not exceed three times, and the plate material is not considered. The average effective utilization rate can reach 89.29%. But the disadvantage is that it is only applicable to the packing problem under the above conditions, lacking generalization.

References

[1] J Y Feng, Y Liu 2022 Optimization of Evolutionary Genetic Algorithm for Rectangular Nesting Grading for Utilization[J]. *Mechanical Design and Manufacture*:1-6*(in Chinese)*.

[2] W P Tang, K Wang, X Huang 2021 A Hybrid Genetic Algorithm for Two Dimensional Orthogonal Layout of Rectangular Parts [J]. Forging and Stamping Technology, 46(10):106-111 *(in Chinese)*.

[3] H Rao 2019 Solution and Application of Plate Cutting Problem [D]. *Jiangxi University of Finance and Economics (in Chinese).*

[4] Q L Chen 2016 Research on Optimization Algorithm of Three Stage Layout Scheme for Two Dimensional Cutting Stock Problem [D]. *South China University of Technology (in Chinese)*.

[5] Y M Wu, N Cao, F Y Li, L Li 2022 Research on the optimization algorithm of the plate porous processing path based on greedy algorithm [J]. *Journal of Hefei University of Technology(Natural Science)*,45(06):742-745+759 *(in Chinese)* .

[6] X Chen, F Deng, Q S Yan, Y J Qv 2021 An Improved Texture Layout Method Based on Greedy Algorithm [J]. *Journal of Geomatics*,46(S1):253-256 *(in Chinese)*.