Study on Optimization of Three-Stage Sheet Nesting with Integration of Virtual Component Construction and Greedy Algorithm

Yizhe Wang [1], Chaojie Han [2], Guan Zhang [3], Qiang He [1, *]

[1] School of Management, Tianjin University of Traditional Chinese Medicine, Tianjin 301617, China

[2] Huawei Technologies Co., Ltd., Shanghai 201206, China

[3] Tongfang Knowledge Network Digital Technology Co., Ltd., Beijing 100083, China

* Corresponding Author

Abstract. Nesting optimization problems are critical in the industrial manufacturing sector, playing a significant role in cost control and resource conservation. This paper focuses on the nesting problem presented in the first question of the 2022 “China Optics Valley Huawei Cup” Postgraduate Mathematical Modeling Competition, B Question. A three-stage nesting model based on a greedy algorithm is constructed. The specified cutting method is “flush cutting” with a maximum of three stages and consistent cutting line directions within the same stage. Factors such as saw blade width are not considered. By concatenating products with the same width or length into virtual components and processing products in descending order, the algorithm prioritizes placing the longest product available at the bottom left corner of the original sheet, updating the nesting space in stages. Experiments on datasets A1-A4 demonstrate that this algorithm achieves a material utilization rate exceeding 94%, providing an efficient solution for two-dimensional nesting optimization.

Key words: Nesting optimization; Greedy strategy; Material utilization; Three-stage cutting.

1. Introduction

Nesting optimization problems, also known as cutting and filling problems, are crucial in the industrial sector, particularly in manufacturing. [1] They have significant implications for cost control and resource conservation. The essence of nesting problems lies in achieving the optimal solution of the objective function under limited space or resource constraints. The goal of nesting optimization is to improve material utilization and reduce resource waste while simplifying the cutting process.

In the research on rectangular part nesting problems, greedy algorithms are widely used in sorting and positioning optimization scenarios due to their local optimal decision-making characteristics. Feng Jianyun proposed a hierarchical evolutionary genetic algorithm, combining a greedy strategy to optimize sorting, and utilized a hybrid positioning algorithm to improve material utilization and reduce nesting time. [2] Qin Zhenhao adopted a multi-population genetic algorithm and a remaining rectangle matching algorithm, converting irregular parts into rectangular envelopes and using a greedy approach to select the optimal remaining rectangle for nesting. [3] Ma Yingjun constructed a two-stage genetic algorithm combined with a greedy strategy model, achieving product and strip remnant reuse through a greedy strategy. [4] Chen Yeyi proposed a greedy hybrid positioning algorithm for three-stage nesting, improving material utilization and shortening running time through problem segmentation, partition optimization sorting, and hybrid positioning. [5] Zhang Xin improved the sparrow search algorithm, introducing Tent chaotic mapping and Metropolis criteria, and combined a greedy strategy to optimize rectangular part sorting, improving raw material utilization and reducing nesting time. [6]

Existing research, while extensive, lacks efficient solutions for three-stage nesting problems. This paper builds a model based on a greedy algorithm to solve the nesting problem in the first question of the 2022 “China Optics Valley Huawei Cup” Postgraduate Mathematical Modeling Competition B Question. It aims to provide a more optimized solution for two-dimensional nesting optimization in actual production.

2. Nesting Problem Modeling Based on Greedy Algorithm

2.1 Problem Description

This study focuses on the sheet cutting problem in Question 1 of Problem B of the 2022 "Optics Valley of China · Huawei Cup" Graduate Mathematical Modeling Competition. The cutting method is restricted to "straight cutting" — specifically, cutting along a straight line perpendicular to one side of the sheet, with each cut necessarily separating the sheet into two pieces. The cutting process can be divided into a maximum of 3 stages, and all cutting lines in the same stage must maintain a consistent direction. Raw sheets are of uniform specifications and sufficient quantity. When designing the nesting scheme, the impact of saw kerf width does not need to be considered. The final products must be intact, and splicing is not allowed for production. Figure 1 shows a cutting example.

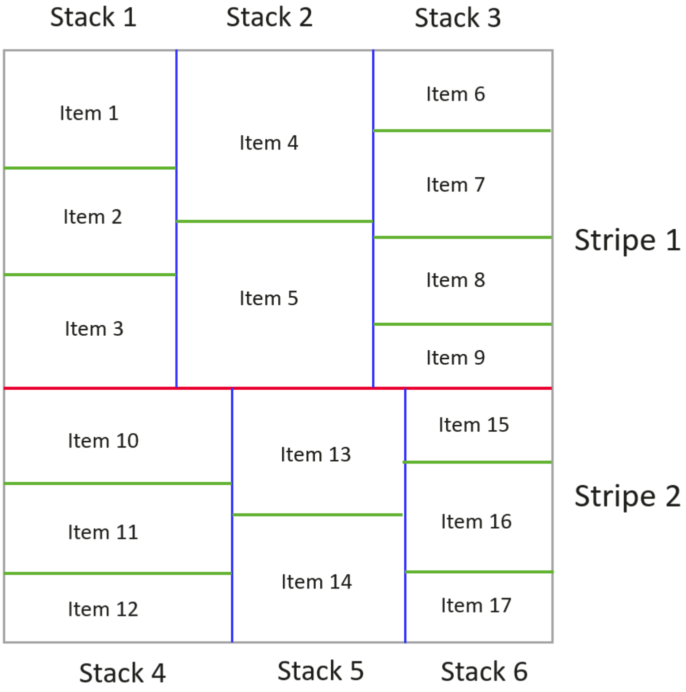

Figure 1. Schematic Diagram of the Three-Stage Cutting Method and Its Generated Modules

2.2 Model Assumptions and Symbol Description

The model assumptions are as follows:

The saw blade width is ignored during the nesting process, allowing for tight arrangement of all product items.

The dimensional data of the product items is accurate and error-free.

The processing order of the products does not affect the nesting results. The symbols used in the paper are explained in Table 1:

Table 1. Symbol Description

Symbol	Meaning
L	Length of the original sheet
W	Width of the original sheet
m	Number of product types
Pi	The i-th nesting method
xi	Number of times the i-th nesting method is selected
aij	Number of the j-th product in the i-th nesting method
M	Number of original sheets used
Q	Set of products not involved in nesting

2.3 Data Preprocessing

According to the greedy algorithm, the longest item is prioritized for placement on the original sheet. Consider concatenating products with the same width or length into virtual items along the length direction. To control time complexity, only concatenation of up to three items is performed, and a unique identification number is assigned to each virtual item. The dataset is updated and a hash table is created to store the corresponding relationship between the original item and the virtual item. After updating the dataset, to achieve a more reasonable and efficient solution algorithm, the products are processed in descending order based on length and width, and to facilitate the “rotation” operation in the algorithm implementation, the matching of length and width is not limited to the definition of length and width in the given data. For example, if the length of item1 is equal to the width of item2, it can also be tightly connected during nesting.

2.4 Model Establishment

Assume the length of the original sheet is L and the width is D. First, consider a single original sheet as a two-dimensional plane with boundaries. The edge of the original sheet along the length is set as the x-axis, and the edge along the width is set as the y-axis, with the top vertex as the origin to establish a planar geometric coordinate system, as shown in Figure 2.

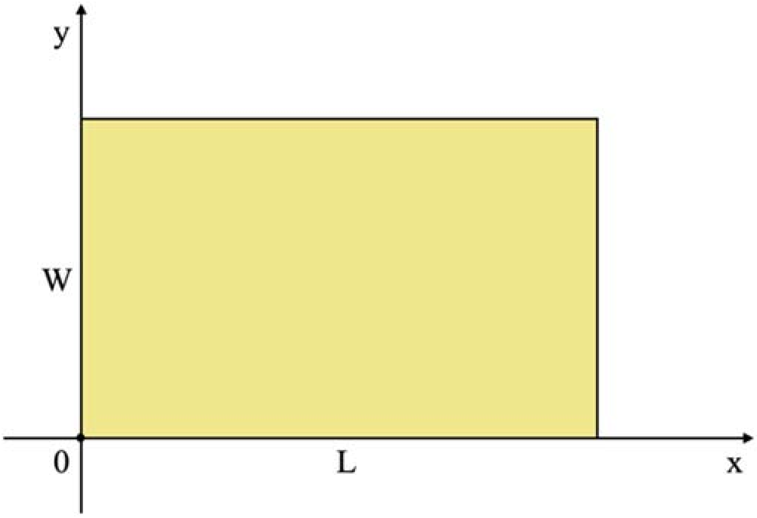

Figure 2. Schematic Diagram of the Coordinate System

Given the product set, let the total number of products be n, the number of product types be m, and the number of different nesting methods on each sheet of the original sheet be k, denoted as P1, P2,⋯, Pk, respectively. The i-th nesting method is selected xi times, where xi∈{0,1,2,⋯}. Simultaneously, the j-th product is aij in the i-th nesting method, where i=1,2,⋯,k, and j=1,2,⋯,m. The optimization objective of the nesting problem is to maximize material utilization. With known products, the optimization objective is transformed into the minimum number of original sheets used. Based on the above settings, the following model is established for the problem:

$$minM = min\sum_{i=1}^{k} x_i \tag{1}$$

$$s.t.\ \sum_{j=1}^{m} a_{ij}\, x_j \geq 1, i = 1,2,\cdots,k \tag{2}$$

2.5 Algorithm Implementation and Solution Analysis

To fully utilize the space of the original sheet, this paper adopts a nesting strategy based on a greedy algorithm, designing specific algorithms based on the above model. Set all items not involved in nesting as a set Q. Since the type of the original sheet is uniform, only the nesting process on a single original sheet needs to be analyzed, and the termination condition of the iteration is that the set Q is empty or no new items can be nested in the original sheet under the requirements of the algorithm. Set iteration parameters Lp and Wp, representing the size of the nesting space S. During the nesting process, consider the rotation of all items, and set the length of the item parallel to the x-axis as li and the width parallel to the y-axis as wi. Therefore, in the process of nesting search, there is no strict requirement that li≥wi.

For a new original sheet, set Lp=L, Wp=W, and the nesting space is the entire original sheet. Traverse the search in Q for items that satisfy li≤Lp, Wi≤Wp, and are as long as possible in the x-axis direction, and place them at the bottom left corner of the original sheet, as shown in Figure 3. At this time, li is used as the base length of a stripe, and the nesting space S and iteration parameters Lp=L, Wp=W-w1 are updated. Update the set Q, delete the item and all items related to it, ensuring that only one original sheet is used once. For example, if the current item placed is item1, then item1 itself and all virtual items containing item1 are deleted in set Q.

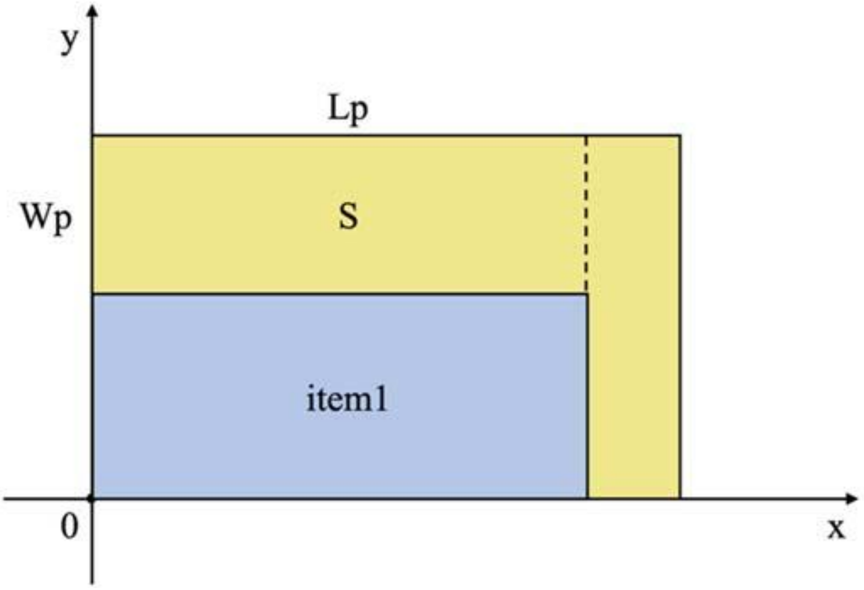

Figure 3. Schematic Diagram of the First-Stage Nesting

In the new nesting space, traverse the search in Q for items that satisfy li≤Lp, Wi≤Wp, and are as long as possible in the x-axis direction, and place them at the bottom left corner of S. At this time, there are two considerations for updating the nesting space: the first is to update along the y-axis direction of the existing item, as shown in Figure 4, at this time the parameters are updated to Lp=L-l1-l2, Wp=W-w1-w2, continue to update the set Q, and then you can continue to perform this nesting strategy in S; the second is to update the nesting space along the x-axis direction of the existing item, as shown in Figure 5, at this time the parameters are updated to Lp=l1-l2, Wp=w2, update Q and traverse the search in it for items that satisfy li≤Lp, Wi=Wp (here is the guarantee of precise cutting of the three stages) and are as long as possible in the x-axis direction, and place them at the bottom left corner of S. If no items that meet the search conditions can be nested in the current stripe, the stripe is considered as “full,” and the current iteration step ends. It needs to be specially stated that in the face of these two considerations, the space update in the x-axis direction is given priority. When Q has no items that meet the search conditions and can be nested in the current stripe, the stripe is considered as “full,” and the current iteration step ends.

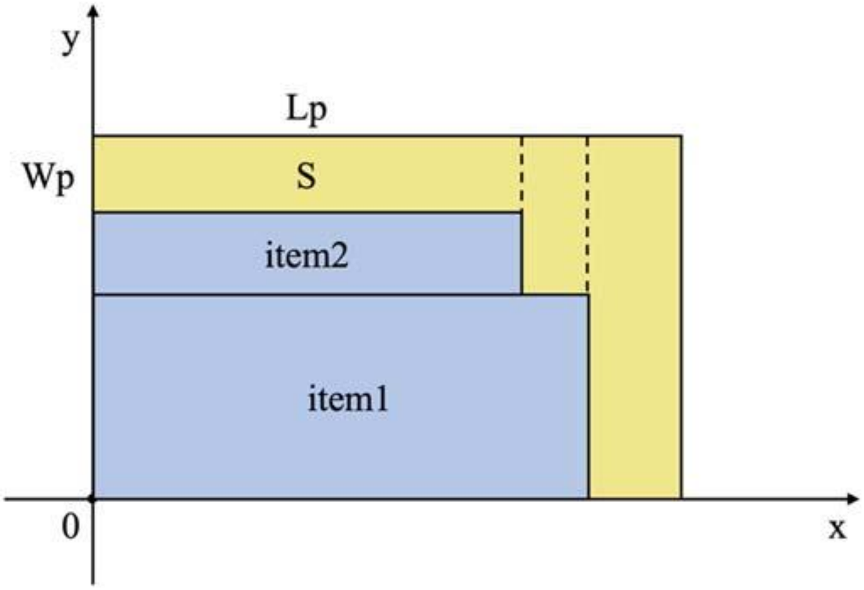

Figure 4. Schematic Diagram of Case 1 in the Second-Stage Nesting

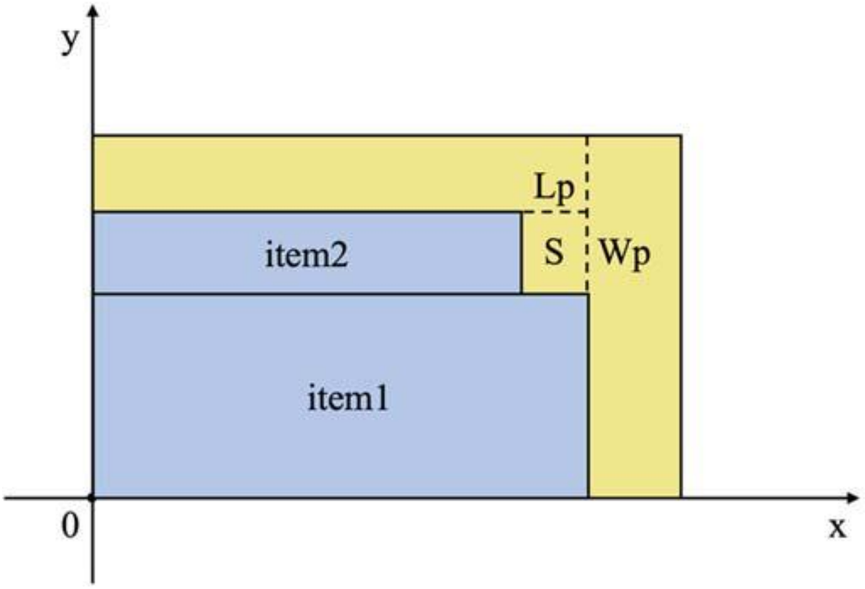

Figure 5. Schematic Diagram of Case 2 in the Second-Stage Nesting

In the remaining part of the x-axis direction of the stripe that has been filled, update it as the nesting space S, as shown in Figure 6, and update the iteration parameters Lp=L-l1, Wp=W at the same time. Update Q and traverse the search in Q for items that satisfy li≤Lp, Wi≤Wp and are as long as possible in the x-axis direction. If there is none, it indicates that the nesting of the original sheet is complete. If Q is not empty, then select a new original sheet to start a new algorithm iteration; if there is, place it at the bottom left corner of S, update the iteration parameters Lp=L-l1, Wp=W, and in this way, the length of the base of the second stripe is limited. Then repeat the operation of step 2 until this stripe is also filled, and repeat this step operation.

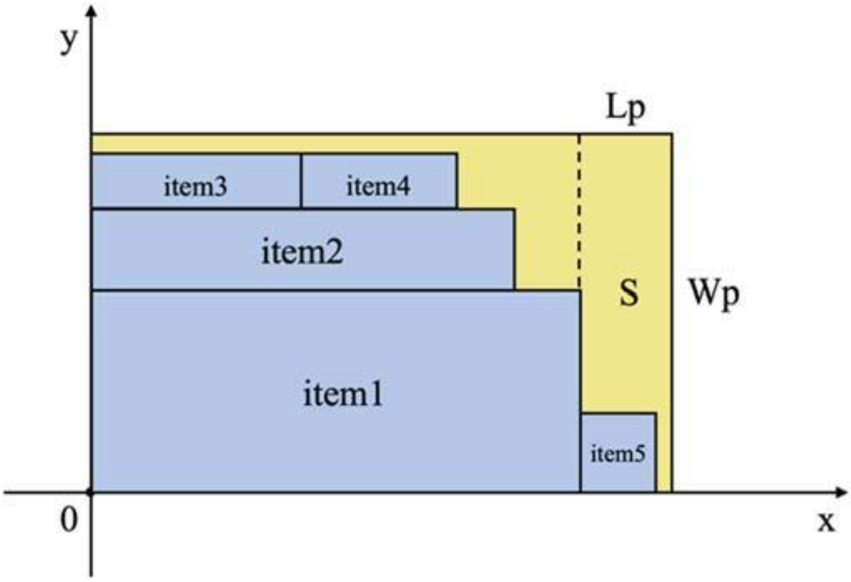

Figure 6. Schematic Diagram of the Third-Stage Nesting

Based on data preprocessing, the above algorithm is used for nesting for the known datasets A1, A2, A3, and A4 in question 1. The results are listed in detail in the table below. In the four test datasets, the algorithm in this paper can achieve a material utilization rate of more than 94%, with excellent running time and good practical application value.

Table 2. Results of the Problem

Dataset	Number of Original Sheets	Material Utilization
A1	88	94.9332%
A2	88	94.1753%
A3	88	95.1467%
A4	86	95.1778%

3. Conclusion

This paper focuses on the nesting problem in the first question of the 2022 “China Optics Valley Huawei Cup” Postgraduate Mathematical Modeling Competition B Question and constructs a three-stage nesting model based on a greedy algorithm, providing an innovative solution for two-dimensional nesting optimization. The research concatenates products with the same width or length into virtual components to achieve optimized product preprocessing, effectively reducing nesting complexity. At the same time, products are processed in descending order, laying the data foundation for the implementation of the greedy strategy. The design of prioritizing the placement of the longest product available at the bottom left corner of the original sheet and updating the nesting space in stages ensures the efficiency of the nesting process and the maximization of space utilization. After experimental verification on datasets A1-A4, the algorithm demonstrates excellent performance. The material utilization rate reaches over 94%, with the utilization rate of dataset A4 reaching as high as 95.1778%. This data fully demonstrates the significant effect of the model in terms of resource conservation.

The successful construction of this model provides new ideas and methods for two-dimensional nesting optimization in industrial production, having important guiding significance for cost control and resource conservation. It not only solves the nesting problem of a specific competition question but also provides a reference paradigm for similar industrial nesting scenarios. However, we are also aware that the adaptability and efficiency of the algorithm still have room for improvement when facing more complex nesting scenarios or large-scale data. For example, when there are many types of products and large size differences, the local optimal decision-making of the greedy strategy may not guarantee global optimality; when processing large-scale data, the time complexity of the algorithm may need to be further optimized. In the future, we will conduct in-depth research on these shortcomings to explore better algorithm strategies and improve the performance of the model in complex scenarios, promoting the wider application of nesting optimization technology in industrial production.

References

[1] Wang, L. (2021). Genetic simulated annealing hybrid algorithm for rectangular part nesting problems. Forging Technology, 46(08), 70-76.

[2] Feng, J. Y., & Liu, Y. (2022). Graded evolutionary genetic algorithm optimization for rectangular nesting based on utilization. Mechanical Design and Manufacturing, (11), 11-16.

[3] Qin, Z. H. (2022). Optimization nesting of irregular parts based on multi-population genetic algorithm and remaining rectangle matching algorithm. Modern Industrial Economics and Information, 12(12), 262-264.

[4] Ma, Y. J., & Zhong, J. J. (2023). Multi-constraint nesting optimization method combining two-stage genetic algorithm and greedy strategy. Journal of Xiamen University of Technology, 31(03), 79-87.

[5] Chen, Y. Y., & Li, H. D. (2024). Research on three-stage nesting problem based on greedy hybrid positioning algorithm. Machinery & Electronics, 42(03), 12-16+25.

[6] Zhang, X., Li, H. X., & Liang, M. (2024). Research on rectangular part nesting optimization based on improved sparrow search algorithm. Journal of Shandong Jianzhu University, 39(06), 108-116.