Research on Cutting-Stock Method of Profile

ShuiPing Li1, WenHui Cai2,a
1,2 School of Mechanical and Electrical Engineering, Wuhan University of Technology, Wuhan, China
a cwh321@whut.edu.cn

Abstract. There is a lot of waste of profile materials in the production process of aluminium alloy doors and windows manufacturers, and the cost of profile materials occupies a large proportion of the manufacturing cost of aluminium alloy doors and windows. According to the traditional aluminium alloy door and window manufacturers in the cutting-stock process, the problem of waste of profiles, confusion of cutting orders, low automation and low production efficiency, etc., we proposed a cutting-stock algorithm that selects different cutting-stock algorithms for different cutting-stock order sizes. We used improved linear programming method to solve the small-scale cutting problem, and limited the number of cutting methods of the profiles, and used LINGO software code to solve it. The hybrid genetic algorithm is used to solve the one-dimensional large-scale order placement problem. It is concluded that the algorithm improves the utilization rate of the profile, improves the production efficiency of the enterprise and decrease the cost of the enterprise.

1. Introduction
Through the field survey of aluminum alloy door and window manufacturing enterprises, it is found that the cost of profiles occupies a large proportion of the manufacturing cost of aluminum alloy doors and windows, and aluminum alloy door and window manufacturers have a large amount of waste in the production process. The reason is that the company still uses the traditional manual cutting mode in the cutting process. The manual cutting mode is generally to cut the longest component from the profile, or the cutting will be given priority to the high demand according to the demand of the component until the component of any specification can’t be cut. The cut scraps are treated as waste, and the rest will be put aside, and when there is a suitable component specification, the rest will be cut. The manual cutting mode is not only low in production efficiency, but also difficult to ensure efficient use of profiles.

The optimal blanking problem is a typical NP-C (Nondeterministic Polynomial-Complete) problem, where NP refers to non-deterministic polynomials. Parts with different lengths and sizes are combined and cut on a fixed raw material. There are many feasible solutions. It Can’t be sure to find the optimal solution. In order to make the cutting problem can be solved efficiently, people have studied many algorithms to solve the optimal cutting problem.

In 1996, Richard Vahrenkamp[1] proposed a sequential heuristic algorithm to solve the one-dimensional blanking problem in the case of equal-length raw materials. At the same time, a random search method is used to select the cutting method. Since the random search uses a probability distribution, the efficiency of random search is not high when the length of the parts differs greatly.

In 1999, Gradisar M[2][3], and others used a linear programming-based method, combined with cutting methods and order requirements, to form a sequential heuristic algorithm to solve the one-
dimensional blanking problem. At the same time, it is also proposed to give priority to the cutting of parts with high demand. Bret J. Wagner [4] used genetic algorithm to solve the one-dimensional cutting problem of bunch cutting. Chien-Tun Yang [5] uses an improved Tabu Search method to solve the one-dimensional blanking problem.

2. Overview of the cutting process and order size division

2.1. Overview of the cutting process
Blanking refers to the sawing equipment cutting the raw materials to the required size. It is the first processing step in the processing of doors and windows. Before blanking, the various dimensions to be cut out of the profile are calculated, that is, the blanking plan is solved. The sawing equipment then saws the profile into door and window components according to the set blanking plan. Then the components are transported to the equipment in the subsequent process, and the components are waiting to be proceed in the next step. In aluminum alloy doors and windows, there are many types and quantities of profiles that need to be cut and sawed. When manually solving the blanking plan, the workload is huge and errors are prone to occur.

![Figure 1. Aluminium alloy profile](image)

2.2. Order size division
After the blanking workshop receives the blanking order, combined with the demand of the overall order, the total demand of a single component can be calculated. The problem of optimized blanking has high NP complexity, and there are many types of aluminum alloy doors and windows, and the size of blanking orders is different. For different blanking data structures, a single optimized blanking algorithm is not universal and cannot be targeted. Solve. If only one optimized blanking algorithm is used, it is difficult to find a good blanking plan when faced with various types of blanking data. There are many kinds of component combinations for different blanking orders, and the data structure composed of its length is complex and diverse. Especially for large-scale orders, in the blanking list of profile BGY4501, the shortest component length is 524mm, the longest is 1453mm, there are 15 types of components, and the prototype length is 6000mm. If the traditional blanking method is used, find out first for all the cutting methods of profiles, there are thousands of combinations of components on the prototype, and it is quite difficult to solve the cutting methods. If the order size can be divided according to the number of component types, and different algorithms are used to solve them separately, the solution algorithm for small-scale order applications can quickly solve the optimal blanking plan, and the number of cutting methods is as small as possible; large-scale orders can avoid cutting methods the solution, and the solution time is short.

Based on the above reasons, this article proposes to use different algorithms to solve different order sizes.

The order size is divided and took into account the actual daily output of the door and window factory, a shift (4 hours) can only produce up to 20 different components, and the total number of components is about 800. Door and window orders are divided into large-scale orders and small-scale orders according to the length and size of the components. The specific classification criteria are...
shown in Table 1. Among them, large-scale orders not only have more types of components, but also have a larger total number of components.

Table 1. Order size division.

| Type of component M | Component size         |
|--------------------|------------------------|
| Small-scale order  | \( 0 \leq M < 10 \)    |
|                    | Nearly half of the profile |
| Large-scale order  | \( 10 \leq M \)        |
|                    | Long or short           |

3. Small-scale one-dimensional cutting problem and its solution

3.1. Small-scale blanking problem description

In the process of blanking, changing the cutting method requires adjustment of equipment and cutters, and small-scale orders have fewer types and quantities of door and window components, so a single cutting method is used very few times, and some may have to be replaced once it is applied. Cutting method. Frequent switching of cutting methods will consume a lot of time and cost, equipment and tools will also be worn to an indeterminate degree, and the number of materials cut each day will also be limited, resulting in components that cannot be delivered on a determined date. And when there are too many cutting methods for the actual problem, the calculation will become very complicated. Therefore, the goal of solving the small-scale blanking problem is to consume the least raw materials while completing the blanking task on time, and at the same time, the number of cutting methods is as few as possible. Based on the above analysis, this paper proposes a linear programming solution method that limits the number of cutting methods, and uses LINGO11.0 software to solve the optimal cutting plan for profiles.

Suppose the quantity of the k-th profile in a certain cutting method is \( d_k \) (\( d_k \) is integer). Then the mathematical model of this cutting method is \( l_1d_1 + l_2d_2 + \cdots + l_kd_k \leq L, \) that is \( \sum l_kd_k \leq L, k = 1,2, \ldots, n. \) Assuming that there are \( m \) different profile cutting methods, combined with the demand for components in the order, the number of applications of each cutting method can be found. \( a_{ij} \) is the number that the i-th component can be cut under the j-th cutting method, it can be obtained that the quantity of the first component cut in the first cutting method is \( a_{11}. \) \( a_{jm} \) represents the quantity of the j-th component produced under the m-th cutting method, and we have established the following relationship matrix for this corresponding relationship.

\[
A = \begin{bmatrix}
    a_{11} & a_{12} & \cdots & a_{1m} \\
    a_{21} & a_{22} & \cdots & a_{2m} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{j1} & a_{j2} & \cdots & a_{jm}
\end{bmatrix}
\]  

(1)

Add a decision variable \( a_{ij} (i = 1,2,3, j = 1,2,3,4) \) indicates the number of j-th parts that can be cut in the i-th cutting method. The goal is to consume the least raw materials; the constraint condition is that the number of cut components meets the demand of the order, and the remaining material cut is smaller than the minimum size of the components, so the non-linear programming model 2 established is as follows:

\[
\begin{align*}
\text{s.t.} & \quad \sum_{i=1}^{m} a_{ij}x_i \geq d_j, j = 1,2, \ldots, n \\
& \quad L - \sum_{j=1}^{n} l_j a_{ij} \leq \min l_j, i = 1,2, \ldots, m \\
& \quad x_i \geq 0, a_{ij} \geq 0, i = 1,2, \ldots, m, j = 1,2, \ldots, n
\end{align*}
\]  

(2)

Where \( n \) is the type of component, \( m \) is the number of cutting methods, \( l_i \) is the length of the member, \( d_i \) is the required quantity of the component, \( \min l_j \) is the smallest size in the component.
According to the previous order, the number of cutting methods is limited to 5, which is solved by LINGO 11.0 software.

After 55398 iterations of calculations, the obtained cutting optimization plan is: plan 1 cuts 20 roots, plan 3 cuts 38 roots, plan 4 cuts 9 roots, plan 7 cuts 2 roots, and plan 8 cuts 4 roots. A total of 73 raw materials are consumed. The total waste length is 11550mm, the raw material utilization rate is 97.36%, and the number of cut components meets the order requirements. The utilization rate of model 1 is basically the same, but the solution of model 2 does not need to find all the cutting methods first. The number of cutting methods can be reduced as much as possible, the workload is reduced, and the work efficiency of the blanking workshop is improved.

4. Large-scale one-dimensional cutting problem and its solution

4.1. Large-scale blanking problem description

4.2. Hybrid genetic algorithm to solve large-scale one-dimensional cutting problem

For the large-scale one-dimensional cutting problem, this paper uses an improved genetic algorithm to solve the problem. It uses real number coding based on component sorting without solving the cutting method. At the same time, a heuristic rule of "selective insertion" is constructed. In the genetic algorithm, the local search mechanism based on is added to the framework to strengthen the local search ability, which has great significance for improving the solution quality of optimizing the cutting algorithm.

Using a decimal coding scheme based on construction ordering, all construction serial numbers are randomly arranged as an individual, and the length is the sum of the numbers of all the components. The arrangement of the serial numbers also directly reflects the cutting order of the components on the raw materials.

When evaluating the individual and output blanking plan, it is necessary to decode according to the length of the raw material and the size of the component. According to the individual code from left to right, the components are cut on the raw materials one by one. Only when the remaining material on a certain raw material is not enough for the component size, it will be transferred to the next one, and the cutting will continue until all components have been cut. For example (1322|311|42) as a cutting plan, it means to cut component 1, component 3, component 2, and component 2 on the first raw material; when cutting the next component 3, if the remaining material is insufficient, it will be transferred to the next one is to cut the component 3 and the two components 1 on the second raw material; when the remaining material is insufficient when the component 4 is cut, it will be transferred to the third one, and the components 4 and 2 are cut. This solution uses 3 raw materials.

Randomly generated $\sum_{i=1}^{m} d_i$ numbers. For example (4444|8800|5560), the number in brackets is the length of the part, and the sum of the number is the required quantity of the component.

The final optimization goal of the large-scale one-dimensional blanking problem studied in this paper is to minimize the number of raw materials consumed and the longest remaining material on the last raw material, so the individual fitness function is set as:

$$f(p) = l_m/[k \cdot L - Q_k]$$

(3)

in the formula, $l_m = \sum_{i=1}^{m} l_i d_i$. That is, the total length of all parts when the blanking is completed, L is the length of the raw material, and k is the number of raw materials consumed in the blanking plan, $Q_k$ is the remaining material on the last raw material, the maximum value of this formula is 1, that is, the raw material utilization rate reaches 100%.

In this paper, the linear variation rate is used. The linear variation formula is $R = [(0.8G - g)/G] + 0.1$, where R is the mutation rate of the current algebra, G represents the total algebra that needs to be evolved, and g represents the current algebra. It can be seen from the formula that the mutation rate decreases continuously with the increase of evolutionary algebra, ranging from 0.1 to 0.9.
In order to speed up the convergence and obtain the optimal solution, a local search mechanism is added to the genetic algorithm framework to form an optimization algorithm that combines local search and global search.

For solving the one-dimensional blanking problem, our ultimate goal is to solve the blanking scheme that consumes less raw materials, and at the same time, the material utilization rate is also greater. Based on this, this paper constructs a "selective insertion" heuristic algorithm for local search. The core idea is: for an individual (cutting plan) in the genetic algorithm, try to minimize the waste of each raw material. For the first raw material in the blanking plan, it is first necessary to judge whether the remaining material after cutting the raw material is greater than the minimum component length, and then search for the following components to see if they meet the queue-inserting conditions, if so, insert the component into this raw material end. Use the same search method to operate each of the following materials in turn until the last one. In this way, the previous cutting plan can be improved to the local optimum.

4.3. Case calculation and analysis
Use the genetic algorithm developed in this chapter to solve large-scale orders and small-scale orders respectively, and compare and analyze the solution solutions.

A large-scale order from the actual production of door and window enterprises, optimized cutting for 45 window sashes, the profile numbers used for cutting are BGY4505, the length of the profiles are 6000mm, and the number is unlimited. The specific component information of component 45 window sash is shown in Table 2, and the genetic algorithm convergence diagram is shown in Figure 2:

| Component number | Length of component | Numbers of component |
|------------------|---------------------|----------------------|
| 1                | 524                 | 6                    |
| 2                | 1252.5              | 8                    |
| 3                | 1253                | 2                    |
| 4                | 560.5               | 4                    |
| 5                | 531                 | 6                    |
| 6                | 1162.5              | 2                    |
| 7                | 790                 | 2                    |
| 8                | 460.5               | 4                    |
| 9                | 1173                | 4                    |
| 10               | 611                 | 8                    |
| 11               | 1453                | 4                    |
| 12               | 811                 | 4                    |
| 13               | 1223                | 2                    |

The above-mentioned large-scale order of 45 window sashes is solved by genetic algorithm, and the optimized solution obtained consumes 9 profiles and has 9 different cutting methods. The solution takes 23.648 seconds and the last remaining material is 5079mm, because the last remaining material will be reused, so the total utilization rate after removing the last remaining material is 97.968%.

The convergence diagram of this genetic algorithm solution is shown in Figure 2.
Figure 2. Convergence graph of genetic algorithm solution

5. Conclusion
With the one-dimensional cutting-stock problem as the background, through studied the Aluminium Alloys doors and windows production process and the cutting-stock process, and combined with the enterprise orders and production points, author came up with Cutting-Stock algorithms which based on the door and window order scales to improve the production efficiency of relevant enterprise.

An optimized linear programming algorithm is proposed for small-scale order placement. This algorithm can increase the average utilization rate of profiles to 97.3% with fewer cutting methods.

An improved adaptive genetic algorithm is proposed for large-scale placing orders, which can efficiently solve large-scale one-dimensional placing orders, and the average utilization rate of profiles is above 96%.

References
[1] Vahrenkamp R. Random search in the one-dimensional cutting stock problem[J]. European Journal of Operational Research, 1996, 95(1):191-200.
[2] Gradisar M, Kljajic M, Resinovic G, et al. A sequential heuristic procedure for one-dimensional cutting[J]. European Journal of Operational Research, 1999, 114(3):557-568.
[3] GradisAr M, Resinovic G, Kljajic M. A hybrid approach for optimization of one-dimensional cutting[J]. European Journal of Operational Research, 1999, 119(3):719-728.
[4] Wagner B J. A genetic algorithm solution for one-dimensional bundled stock cutting[J]. European Journal of Operational Research, 1999, 117(2):368-381.
[5] Yang C T, Sung T C, Weng W C. An improved tabu search approach with mixed objective function for one-dimensional cutting stock problems[J]. Advances in Engineering Software, 2006, 37(8):502-513.