Pallet Loading Problem Model Based on Nested Loop Algorithm

Ruiqi Zhan

School of Electrical & Electronic Engineering, North China Electric Power University, Baoding 071000, China.

739131665@qq.com

Key Words: integer programming, two-dimensional layout, rectangular layout, area constraint

Abstract. This model is based on the problem B in the *Mathematical Modeling Contest of North China Electronic Power University* in 2015. It works out solutions for non-overlapping box arrangement with load-bearing limits and without bearing limits, which can finally solve the practical loading problem of overlapping of multi-type boxes considering bearing limits.

About the mentioned practical problem, it is actually a two-dimensional layout problem. In this paper, models for integer layout in unified direction are established by integer programming. First, select an area-undetermined rectangle in the lower-left corner of the pallet and pack boxes X in it in unified direction. Then divide the residual area into two pars, B and C, and pack boxes Y in each part in unified direction. The layout of the residual area is concerned with the size of the lower-left corner. To reduce discussions, the maximum area utilization of B and C is set as constraint which will be considered comprehensively, the minimum of the whole oddments of three areas be set as objective function, thus four layout methods can be obtained. Then exchange the box types in the selected rectangle and the residual area, there comes another four layout methods. By analyzing the packing and division methods of the eight layout, the optimal solution can be concluded.

Introduction

As there are some rectangular boxes (see the size and weight details in Annex 1)need to be packed, the 1210 pallet (1200mm×1000mm) is considered to be adapted, which is used widely currently. Then we search for a layout solution for packing Box 1 transversely and non-overlapped on a 1210 pallet, making the area utilization as high as possible. This paper builds a model and obtain a layout solution with maximum utilization by packing two arbitrary specifications of boxes on a pallet of arbitrary size.

By integer programming, a rectangular area of size-undetermined be first selected in the lower-left corner of the pallet, and boxes X be packed unified in transverse or vertical direction in it. Then, transversely or vertically divide the residual area into two parts and pack boxes Y in each part. Similarly, boxes in each area are in unified direction. As the size of the residual area is relevant to the size of the lower-left rectangle, the box layout in the residual area is concerned with the size, so the utilization area of the residual part is set as constraint, the minimum of the whole oddments of the three area as objective function, integer solutions under eight circumstances can be obtained.

When exchange the box types on the pallet with boxes X in residual area and Y in the lower-left rectangle, and similarly divide the residual area, another eight layout solutions can be gotten. The optimal box layout solution[1] and the minimal oddments will then be obtained by comparing the sixteen solutions.

In order to reduce the computation, the constraint condition is revised to the maximum utilization of the residual area, and which lessen the number of circumstances to eight.

Models for integer layout in unified direction

Modeling. ①Designate a rectangle in the lower-left corner with boxes of same size in unified direction in it.

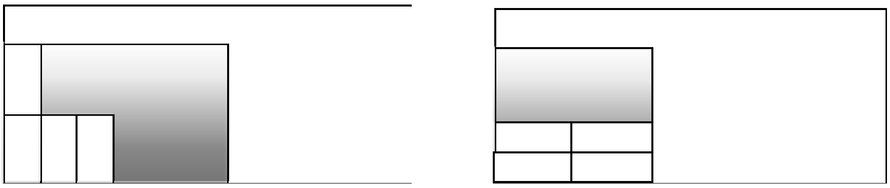

Fig.1: Two ways of box packing

When the boxes in the lower-left rectangle are within the area limitation of the pallet, the box number can be regarded as constraint condition. Packing Box 1 transversely in the lower-left corner is taken as an example for analyzing.

Constraint condition:

$$\begin{cases} 1 \le M \le \dfrac{1200}{a_1} \\ 1 \le N \le \dfrac{1000}{b_1} \end{cases} \quad (1)$$

Note: In this formula, M is the number of the long edge of the lower-left rectangle with which can pack the most *Box 1* along. N is the number of the short edge of the lower-left rectangle that can pack the most *Box 1*.

②There are four methods to divide the residual area into two parts by transverse or vertical cut, then pack each part with the other type of box respectively.

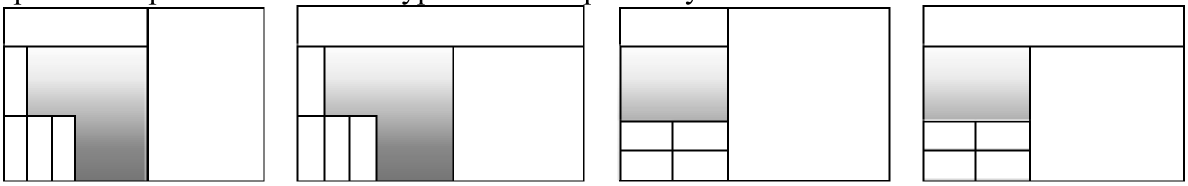

Fig.2:Four ways of area division

③Pack each residual area with the other type of boxes unified in direction, and eight situations will be resulted in. Two of them are illustrated as follows. B

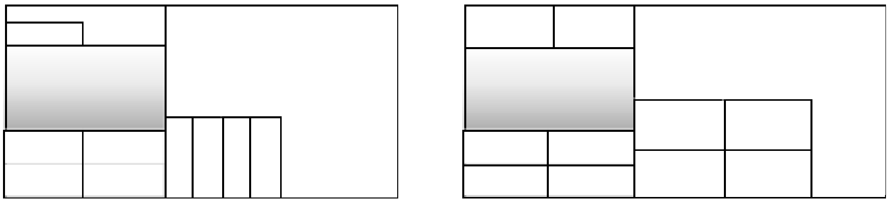

Fig.3: Box layout methods: four methods for each division, and eight in total

Define the maximum utilization area of part B and C as the constraint condition. For each area, the concerning issue is the layout of Box 5, making the maximum utilization area as constraint can reduce the discussions.

$$S_B = \max([1200/a_5] \times [(1000 - Nb_1)/b_5]a_5b_5, [1200/b_5] \times [(1000 - Nb_1)/a_5]a_5b_5)$$

$$S_C = \max([(1200 - Ma_1)/a_5] \times [Nb_1/b_5]a_5b_5, [(1200 - Ma_1)/b_5] \times [Nb_1/a_5]a_5b_5) \quad (4)$$

Note: [1200/x] indicates the greatest integer not exceeding 1200/x. Some integer programming formulas are listed as follows by taking the area B and C as examples which will be packed with *Box 5* and divided horizontally.

④Objective function for the integer programming

As required in integer programming, the minimal overall oddment is to be the objective function.

$$g_1 = \min(1200 * 1000 - NMa_1b_1 - S_B - S_C) \quad (5)$$

So the integer programming formula comes :

$$g_1 = \min(1200 \times 1000 - NMa_1b_1 - S_B - S_C)$$

$$s.t.\begin{cases} 1 \le M \le \dfrac{1200}{a_1} \\ 1 \le N \le \dfrac{1000}{b_1} \\ S_B = \max([1200 / a_5] \times [(1000 - Nb_1) / b_5]a_5b_5, [1200 / b_5] \times [(1000 - Mb_1) / a_5]a_5b_5) \\ S_C = \max([(1200 - Ma_1) / a_5] \times [Nb_1 / b_5]a_5b_5, [(1200 - Ma_1) / b_5] \times [Nb_1 / a_5]a_5b_5) \end{cases}$$

If the area division method altered, the formula for the maximum area will also change. Thus, the integer programming formulas with area vertically divided can be obtained similarly.

Model Solving. The oddment, box packing methods and row&column number of lower-left rectangle and the residual area in eight circumstances are listed in *Table 1*.

Table 1: The oddments, box packing methods and row&column number of lower-left rectangle and the residual area in eight circumstances

No.	oddment	Box type in lower-left rectangle	Packing method in lower-left rectangle	Number of rows/columns in lower-left rectangle	Box type in residual area/division method	Packing methods in residual area B/C	Number of remaining part B//C
1	21%	1	transverse	2/1	5/transverse cut	transverse/transverse	9//6
2	11%	1	transverse	1/1	5/vertical cut	vertical/vertical	4//15
3	15%	1	vertical	1/1	5/transverse cut	transverse/transverse	12//9
4	11%	1	vertical	1/1	5/vertical cut	vertical/vertical	1//18
5	12%	5	transverse	3/2	1/transverse cut	transverse/vertical	6//2
6	18%	5	transverse	1/1	1/vertical cut	vertical/vertical	2//8
7	34%	5	vertical	1/2	1/transverse cut	transverse/transverse	6//2
8	26%	5	vertical	1/2	1/vertical cut	transverse/transverse	1//8

In conclusion, if pack Box 1 in the lower-left corner in transverse direction for a row and a column, and cut the residual area into part B and C, pack four Box 5 in part B in vertical direction and fifteen Box 5 in part C in vertical direction, the minimal oddments are expected to be achieved, that is 11%.

When pack Box 1 in the lower-left corner in vertical direction for a row and a column, then cut the residual area into part B and C, pack one Box 5 in part B in vertical direction and eighteen Box 5 in part C in vertical direction, the total oddments are expected to be 11% , the minimum value.

The packing solutions are shown in the figures below.

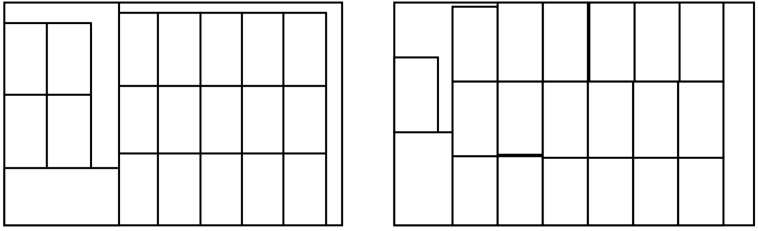

Fig.4: Packing solutions with minimal oddment for *Problem B*

Summary

The difficult point in this problem is the combination of two packing methods, so we adapt algorithms by dividing the area. The whole pallet is divided into three areas, and the maximum utilization of the later two areas is set as the constraint condition of the first area, the minimum value of whole oddments as the objective function.

This model takes eight circumstances into consideration. On the premise that the first area is packed with the same type boxes, transversely or vertically divide the residual area and pack another type of boxes in the later two areas. Then it works out the minimal oddments of the whole eight layout combinations in the three areas by integer programming. It discusses the problem via different circumstances, taking everything into consideration, and obtain a minimal oddment of 11%. All of the above indicates that the model can make the pallet achieve the highest utilization, and it's of good practicability.

References

[1] Li Yong, Application of Approximate Algorithm in Layout Optimization, Huazhong University of Science and Technology, 2005.