An Improved Optimization Algorithm for the Container Loading Problem

Zhu Shunzhi
Department of Computer Science and Technology
Xiamen University of Technology
Xiamen, Fujian 361024, China
szzhu@xmut.edu.cn

Hong Wenxing
Research Centre of System and Control
Xiamen University
Xiamen, Fujian 361005, China
hwx@xmu.edu.cn

***Abstract*—Container loading problem is a kind of space resources optimization problem which consists of various constraints. The solution can be extended to aircraft, cargo loading for ships, even the memory allocation for computer, and other applications. This paper proposes a new algorithm for loading a variety of different goods into a single container with multi-batches. With the concept of "plane" and "block", the algorithm uses "depth priority" strategy to locate for the suitable space. The algorithm also allows goods to rotate in any possible directions, while under the guarantee of efficient space usage, it improves the placement stability. With the priorities of each goods assigned by the algorithm, we should could allocate more goods at the same location. The optimal algorithm is supposed to withdraw when the last batch packing is unsuitable. Experimental results show that the algorithm is effective to solve such problems.**

Keywords--three-dimensional loading; *logistics distribution; heuristic algorithm*

I. INTRODUCTION

Three-dimension loading problem is a layout problem of searching a variety of smaller goods to load into a larger single container. It belongs to cutting and packing problems. The optimization of layout and placement will improve the space usage rate, reduce the cost of transport and increase the profits of the enterprise. The solution of this problem can be also applied to aircraft, cargo shipping, and memory allocation for computer.

In the early sixties, Gilmore and Gomory [6] have studied this problem and proposed some papers and algorithms. Bischoff and Marriott [1] compared several algorithms in their works. Currently, there are still many researchers focusing on the single loading optimization problem. Zhai Yu and Sun X.M. [8] presented a heuristic algorithm for three-dimensional container loading problem with non-identical items. To get a more efficient result, we improve this algorithm, and implement it in Visual C++ language.

The research problem in this paper can be described as follows, for a batch of fixing volume goods, what is the minimum number of containers be allocated; or for a single container, what is the maximum goods can be carried. However, since the real world usually requires the logistics distribution processes to load and unload the goods in several stations, it is necessary to address the multi-loading problem. To resolve this problem, this paper proposes a loading algorithm which consists of (a) the priority of space division and (b) layer optimization after researching in the optimization on multi-loading issues.

II. PROBLEM DEFINITION AND TERMINOLOGY

We can define a multi-loading issue in single container problem as follows. Given a container A with width W, length L and height H and N batch goods Ni with priority, each batch has n kinds of goods in different type denoted as a*1*,a*2*,a*3*,...,a*n*, assume the i*th* (i=1,2,..n) kind of goods has a quantity M*i* with a dimension C*i**K*i**G*i*. The objective is to load these goods subject to their priorities into a single container in order to minimize the wasted place under certain constraints including:

1) Space constraint: Each goods can be put into a container.

2) Box arrangements and stability [5]: The goods have a priori knowledge of balanced positions and can not fall over. This means that additional precautions, e.g. the use of spacing material, are not necessary. In addition, boxes must be arranged completely within the container and parallel to its side walls.

3) Integrity of goods: Put the same kind of goods together.

4) Directional constraint: The goods can be rotated in all six directions, this means, we have six placement methods.

5) Integrity of batch: For the convenience of loading and unloading, each same batch goods should be put together.

III. AN IMPROVED ALGORITHM

Multi-batch loading can be divided as several single loading procedures. When a current loading is in process, there is no priori knowledge of the next batch of goods. Hence, we should reserve a maximum of two-dimension area at the previous loading. The algorithm combines the bottom-up and inner-outer placement methods when we consider the goods placement. First, we use depth priority method to find out the idle space to place goods in the deepest room. Second, every similar goods is packed into different "block" which will be put into the found idle space. Third, in the idle space, a recursive method is utilized from the bottom up, from the inner to outer and from left to right to get a last placement result by computing the optimization efficiently.

978-0-7695-3570-8/09 $25.00 © 2009 IEEE
DOI 10.1109/WCSE.2009.335

A. Basic Approach

1) Goods and container are both cuboids, hence, we can set up a coordinate systems for the container illustrated in Figure.1, where denotes the length (depth of the container) as X, width as Y, height as Z.

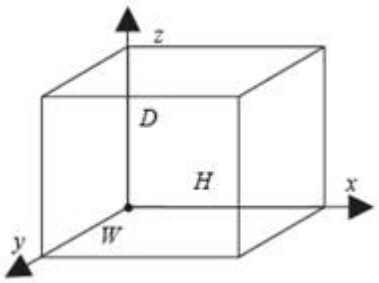

Figure 1. 3-D Coordinates System

2) Each box is located in parallel to the side walls of the container. We can use integer *1-6* to denote six rotational directions of each boxes.

3) "Block" is a structure combined by a same type of goods in the same placement direction, as in Figure.2. The different types of goods or goods in different placement directions are all defined as different blocks.

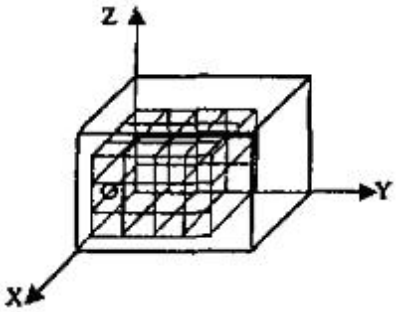

Figure 2. Fill by Blocks

4) The layer line means the maximum value in X-axis when we put the goods boxes from inner to outer, as in Figure.3. Outer from the layer line is a 2-D space, while inner behind the layer line is a 3-D space. [3]

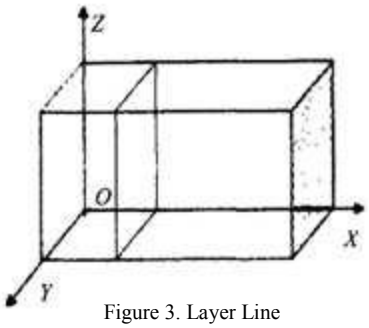

Figure 3. Layer Line

5) In X-axis, we only fulfill a layer of goods, which means optimization in this 2-D space defined above only need to compute the full rate in 2-D Y-Z plane.

6) Comparing to the X-axis optimization, we should compute the full rate of goods in 3-D XYZ space. If the full rate is less than a threshold, the space should be divided into three planes to optimize, i.e., XY, YZ and XZ. According to the rule described in (5), we should choose the best full rate.

B. Choosing Space

At the initiate step, there is only one single plane candidate, i.e., the container. In the following recursive process, choosing a plane should to obey to the following rules:

1) The inner space has the priority, i.e., the plane with the smaller X-axis given the precedence. This is because a worker can not reach the far away plane.

2) The lower space has the priority if they are in the same X, i.e., the plane with the smaller Z-axis given the precedence. As we know, the higher the placement, the greater the instability. To avoid early high pack, we set up this rule.

3) The plane with the smaller area has the priority, if the idle space has the same X and Z axis. A small space is more and more inconvenient to use in the subsequent placement.

4) The longer and narrower plane has the priority, if all the axis and area mentioned above are the same. The same inconvenience to it is in the subsequent placement.

5) If all the idle spaces are the same in the above mentioned four rules, we should compare the Y-axis and select the smallest one.

C. Choose Goods and Determine the Place Direction

After we determine a suitable plane for placement in the last step, we are going to find out the goods and their directions with the largest full rate. Assume that, there are N kinds of goods to be loaded. The idea for this step is that we can set up priority for each goods loading according to the computation result on the volume, weight, density and quantity. Then we can compute each of goods full rate and direction according to the priority. The objective of this step is to find out the goods with the largest full rate, and then determine its direction. Comparing to the volume of single goods, we can select a bigger volume if two goods have the same full rate.

The algorithm to compute the place direction is:

1) For a goods, denote its value of the longest edge, longer edge and shortest edge with a one dimension array a[3], while denote b[6] for a plane with each element corresponding to three value of the longest edge, longer edge and the shortest edge, three labels which edge is the longest one, which one is the 2nd one, and which one is the shortest one.

2) Judge each goods whether they can be put into the space. Base on the notations above, we have five rules here:

a) Neither a[0]>b[0] nor a[1]>b[1] nor a[2]>b[2] , the goods doesn't be fit for the selected place.

b) When b[2]<a[1]b[1]<a[0], there is one suitable place method for the selected place.

c) When a[0]≤b[1] and a[1]≥b[2], there are two methods for placement.

d) When b[2]<a[0]≤b[1] and a[1]≤b[2], there are four methods for placement.

e). When a[0]≤b[2]there are six methods for placement.

3) Under these rules, we define a 2-dimension array c[m][6] to store the placement methods. Here m refers to the quantity of the placement, while the second dimension is a fixed 6 stands for the goods longest edge, the second long edge, shortest edge and which edges in container are in parallel with the longest, the second long edge and the shortest edge of goods correspondingly.

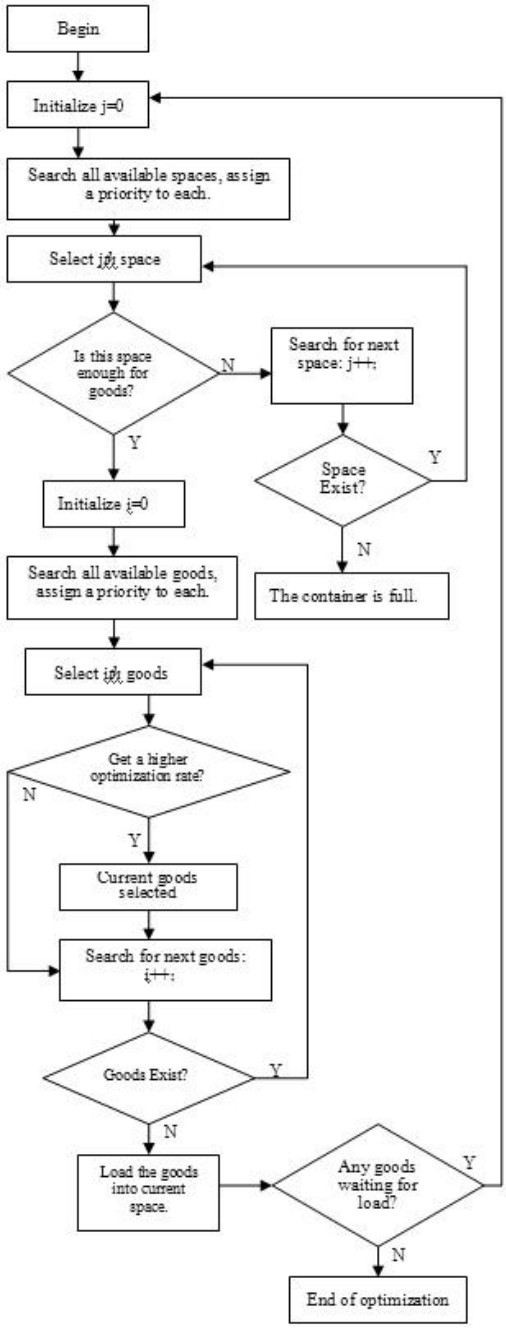

Figure 4. Optimal Flow Chart

D. Generating and Merging the plane

After loading the biggest full rate goods to the plane, there are 0-3 new planes will be produced as shown in Figure.2. To improve the usage on these planes, we can merge small planes to form a bigger one. Intuitively, this operation will be performed under the condition that the newborn plane and its neighbor idle plane are fall in the same Z-axis, and they have an overlay edge.

E. Optimal Flow Chart

Compare to the flow chart addressed in paper [2], we provide the key idea of our algorithm in Figure.4. The left chart illustrates the procedures and steps of determining the placement of the goods.

IV. EXPERIMENT RESULT AND ANALYSIS

We make an experience to evaluate our optimization algorithm which was implemented in Visual C++. A space cut technology presented in [7] and goods placement method introduced in [4] were adopted in our algorithm. We obtained a satisfactory result from the algorithm and the loading optimization is higher than 81%. Assume that the container has a length of 3500 mm, width of 2200 mm and height of 1500mm. Table 1 show the dimensions of goods, while Figure.5 shows the packing result. In this case, we get an optimization rate of 92.08% within 2 seconds.

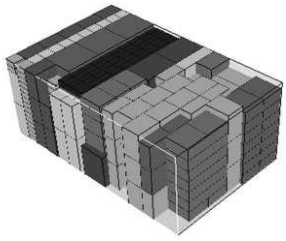

Figure 5. Packing result

TABLE I. DIMENSIONS OF GOODS FOR INSTANCE NUMBER

ID	Length (mm)	Width (mm)	Height (mm)	Color	Number	Load No.
1	200	100	200	Yellow	20	1
2	300	300	400	Grey	15	1
5	500	200	200	Blue	30	1
7	600	540	500	Green	10	1
12	400	500	250	Black	20	1
14	600	500	450	Red	20	1
15	650	250	200	Gold	16	1
17	200	400	450	Green	18	2
3	200	100	100	Blue	60	2
9	400	400	200	Grey	14	2
8	500	200	400	Green	15	2
6	300	400	500	Yellow	30	2

V. CONCLUSIONS

This paper presents an improved heuristic algorithm for multi-batch container loading problem which is able to

obtain an optimal rate of 81%. Comparing to the previous approach introduced in reference [8], this algorithm is more than the optimal rate of container, it has some special constraints such as multi-batch, goods can not fold up, and goods with high density will have a priority in the optimization computation. The experimental result shows that, this algorithm is a useful method for loading problem in logistics. It can be used widely in many scenarios, such as loading multi-types of goods, loading in multi-batch and artificial auxiliary. This makes the loading process more feasible and controllable.

ACKNOWLEDGMENT

This work is partially supported by the Education Department of Fujian Province Funds of China (JA08220).

REFERENCES

[1] E. Bischoff and M. Marriott. A comparative evaluation of heuristics for container loading. *European Journal of Opera tional Research*, 44(2):267–276, 1990.

[2] A. Bortfeldt and H. Gehring. A hybrid genetic algorithm for the container loading problem. *European Journal of Operational Research*, 131(1):143–161, 2001.

[3] Z. Defu, W. Lijun, and C. Qingshan. A combinational heuristic algorithm for the three-dimensional packing problem. *Journal of Software*, 9(18):2083–2089, 2007.

[4] M. Eley. Solving container loading problems by block arrangement. *European Journal of Operational Research*, 141(2):393–409, 2002.

[5] H. Gehring and A. Bortfeldt. A Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operational Research*, 4(5-6):401–418, 1997.

[6] P. Gilmore and R. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120, 1965.

[7] D. Pisinger. Heuristics for the container loading problem. *European Journal of Operational Research*, 141(2):382–392, 2002.

[8] Y. Zhai and S. Xiaoming. A heuristic algorithm for three-dimensional container loading problem with non-identical items. *Journal of Shanghai Jiaotong University*, 41(8):1244–1247, 2007.