Cubage-Weight Balance Algorithm for the Scattered Goods Loading with Two Aims

Liu Xiao-qun¹, Ma Shi-hua², and Li Qi³

¹ National Earthquake Infrastructure Service, Beijing 100036, China

² School of Management, Huazhong University of Sci.&Tec., Wuhan 430074, China

³ Graduate School of Chinese Academy of Sciences, Beijing, 100049, China
{hustlxq, stevenmai, casliqi}@126.com

Abstract. To utilize the volume and carrying capacity to solve the low efficiency of truck loading, a loading planning was made by using optimization methods based on combinatorics. The mathematic model of goods loading was set up and a polynomial approximation algorithm was proposed according to the characteristics of the goods transportation. With the volume and load of the truck taken into consideration, this algorithm can converge at a satisfactory solution and has a good effect in application.

Keywords: loading, optimization, combinatorics, polynomial approximation.

1 Introduction

Loading is a typical 2-Dimensional Packing Problem, which needs to allocate a set of rectangular items to larger rectangularly standardized stock (such as train, steamship and truck) units by minimizing the waste. The truck loading is important because it is reported that the road transportation is of particularly higher transportation costs and lower efficiency when comparing with other transportations. It is known that the utilization of trucks is only about 60% in China. Most algorithms for 2-DPP are heuristic. General reviews on packing and loading problems can be found in Dowsland and *et.al* [1]. Results on the worst-case performance and the probabilistic analysis of packing algorithms can be found in Coffman et al [2].

There are two common approaches for the solution of loading problem. The first approach is based on lower bounds and heuristics based on combinatorial considerations. Chen and Srivastava [3] studied a generalization of the Martello-Toth lower bounding procedure based on a famous lower bound algorithm, and Stephen [4] described a column-generation based on branch-and-bound algorithms. These algorithms are very fast in some cases but not effective enough to provide optimal solutions when embedded within a branch-and-bound scheme.

The second approach is based on an integer programming formulation with a huge number of variables, whose linear programming relaxation can be solved by column generation, typically requiring a considerable time, but obtaining extensive information about the optimal solution of the problem. An integer program was implemented to solve a kind of rectangular packing problem, and Berghammer

developed a linear approximation algorithm for bin packing with absolute approximation factor [5]. One kind of container and multi-container loading problem was solved by Scheithauer [6] based on improved LP. While for bipartite vertex packing problem, strongly polynomial simplex algorithm was introduced by Armstrong [7]. But just as shown in the study of Woeginger [8], we are not aware of any exact algorithm for 2-DPP in the literature even if 2-DPP does not have an asymptotic polynomial time approximation scheme [9] unless $P=NP$.

This article introduces a C-W (short for Cubage-Weight) balance method characteristic by little counters and quick constringency to optimize the solutions mainly based on A_k ¹ and First-Fit² algorithm [10]. It is proved by numerical simulation that C-W balance method can converge to satisfactory solution effectively and has a good effect in practice because it takes into account the truck's volume and carrying capacities at the same time compared with the methods which consider one of two elements preferentially.

2 Model Description, Assumptions and Notation

The problem can be described as follows: there are some trucks in the distribution center. Now scattered goods need to be delivered. How does one make the loading plan rationally in order to utilize the most of truck's cubage and carrying capacity? In this problem, it is assumed that: a) the flow direction of all these scattered goods is the same; b) no goods need to be loaded specially; c) in the process of loading, only the cubage and the carry capacity restrictions are considered. The following additional notation will be used below in the algorithm designing:

$N(i)$: Set of all scattered goods, i is the number of goods i ($i=1,2,\dots,n$),

$N(v_i)$: Volume set of scattered goods, v_i is the volume of goods i ($i \in N(i)$),

$N(g_i)$: Weight set of scattered goods, g_i is the weight of goods i ($i \in N(i)$),

$M(j)$: Set of trucks, j is the number of truck j ($j=1,2,\dots,m$),

$M(V_j)$: Cubage set of trucks, V_j is the cubage of truck j ($j \in M(j)$),

$M(G_j)$: Carry capacity set of trucks, G_j is the carry capacity of truck j ($j \in M(j)$),

x_{ij} : 0-1 Variable, if goods i is loaded into truck j , then $x_{ij} = 1$, else then $x_{ij} = 0$,

y_j : 0-1 Variable, if truck j is selected to loading scattered goods, then $y_j = 1$, else then $y_j = 0$,

$s_j^{(k)}(i)$: Loaded goods set of truck j in the k^{th} iteration of the algorithm, $j \in M(j)$, $i \in N(i)$,

¹ A_k algorithm: Slect goods set s (no more than k elemtens) from $N(i)$, if $\sum_{i \in s} g_i \leq G$ and $\sum_{i \in s} v_i \leq V$, loaded the goods increasingly from s until the truck is full loaded.

² First-Fit algorithm: Seek the most suitable goods for the left truck with the most loading capacity until all the trucks are full loaded. See more in reference [10].

$s_j^*(i)$: Satisfying or optimization solution of truck j , $j \in M(j)$, $i \in N(i)$,

$S^{(k)}$: Loaded goods set of truck set $M(j)$ in the k^{th} iteration of the algorithm,

$S^{(k)} = (s_1^{(k)}(i), s_2^{(k)}(i), \dots, s_m^{(k)}(i))$,

S^* : Satisfying or optimization solution of truck set $M(j)$.

The model of scattered goods loading problem is:

$$\text{Min}Z = \sum_{j=1}^m y_j, \text{Max}Z_g = \sum_{y_j=1} \sum_{i=1}^n g_i x_{ij}, \text{Max}Z_v = \sum_{y_j=1} \sum_{i=1}^n v_i x_{ij}$$

Subject to:

$$\sum_{j=1}^m x_{ij} \leq 1 \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n g_i x_{ij} \leq G_j y_j \quad j = 1, 2, \dots, m$$

$$\sum_{i=1}^n v_i x_{ij} \leq V_j y_j \quad j = 1, 2, \dots, m$$

$$x_{ij} = 0 \text{ or } 1 \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m$$

$$y_j = 0 \text{ or } 1 \quad j = 1, 2, \dots, m$$

The solution is to find goods sets $s_1^{(k)}(i), s_2^{(k)}(i), \dots, s_m^{(k)}(i)$ in $N(i)$ (Here, $s_1^{(k)}(i), s_2^{(k)}(i), \dots, s_m^{(k)}(i)$ should satisfy these conditions: $\forall i, j, s_i^{(k)}(i) \cap s_j^{(k)}(i) = \text{empty set}$, $\bigcup s_i^{(k)}(i) = \bigcup s_j^{(k)}(i) \subseteq N(i)$ ($i \neq j$) in order to realization $\text{Min}Z$, $\text{Max}Z_g$ and

$\text{Max}Z_v$ under the condition of $\forall j \in M, \sum_{i=1}^n g_i x_{ij} \leq G_j y_j$ and $\sum_{i=1}^n v_i x_{ij} \leq V_j y_j$.

3 Method and Algorithm

3.1 One Truck Loading

Defining the notation as follows:

c_i : The ratio of goods volume v_i and weight g_i , $c_i = v_i / g_i$;

C : The ratio of truck's cubage V and weight G , $C = V / G$;

V_{sum} : The total loaded goods volume;

G_{sum} : The total loaded goods weight;

c' : The ratio of V_{sum} and weight G_{sum} , $c' = V_{sum} / G_{sum}$;

r : The truck's utilization rate, $r = c' / C$.

The one truck loading algorithm is as follows: a) compare c_i ($i \in N(i)$) with C , where $N(i)$ is sorted increasingly according to numerical value of $|c_i - C|$ and $(c_i - C) \times (c_{i+1} - C) \leq 0$, $(c_i - C) \times (c_{i+2} - C) \geq 0$, $|c_i - C| \leq |c_{i+1} - C|$, ($i = 1, 2, \dots, n-2$); b) mark the sorted set of $N(i)$ as P ; c) choose the goods from P orderly. If goods can be loaded in the truck, then loads the goods, otherwise the loading process is over. In order not to lost generality, assuming $\max\{g_1, g_2, \dots, g_n\} \leq G$ and $\max\{v_1, v_2, \dots, v_n\} \leq V$ is tenable.

Algorithm of one truck loading

Step 1: Input $N(g_i) = \{g_1, g_2, \dots, g_n\}$, $N(v_i) = \{v_1, v_2, \dots, v_n\}$, G , V .
 Step 2: Calculate $c(i) = \{c_1, c_2, \dots, c_n\}$ ($c_i = v_i / g_i$ ($i = 1, 2, \dots, n$)), $C = V / G$.
 Step 3: Set $P = \text{empty set}$, $V_{sum} = 0$, $G_{sum} = 0$, $c' = 0$, $r = 0$.
 Step 4: Compare C with c_i ($i \in N(i)$), sorted $N(i)$ increasingly according to numerical value of $|c_i - C|$ and $(c_i - C) \times (c_{i+1} - C) \leq 0$, $(c_i - C) \times (c_{i+2} - C) \geq 0$, $|c_i - C| \leq |c_{i+1} - C|$, ($i = 1, 2, \dots, n-2$) and mark the sorted set of $N(i)$ is P .
 Step 5: Supposing $P[1] = i$, if $g_i \leq (G - G_{sum})$ and $v_i \leq (V - V_{sum})$, then run step 6; else $P \setminus P[1] \rightarrow P$, run step 7.
 Step 6: Let $V_{sum} + v_i \rightarrow V_{sum}$, $G_{sum} + g_i \rightarrow G_{sum}$, $c' = V_{sum} / G_{sum}$, $r = c' / C$.
 Step 7: If $P = \text{empty set}$, then run step 8, else run step 5.
 Step 8: Output P , G_{sum} , V_{sum} , c' , r .

3.2 M ($M > 1$) Truck Loading

To M truck loading problem, the heuristic algorithm is designed as follows.

Stage 1: Truck Selection

In this stage, the suited truck is selected according to the most patchable C-W ratio between the total scattered goods and all trucks. Basic function is as follows:

Step 1: Input $N(g_i)$, $N(v_i)$, $M(G_j)$, $M(V_j)$.
 Step 2: Calculate $M(C) = \{C_1, C_2, \dots, C_m\}$ ($C_j = V_j / G_j$ ($j = 1, 2, \dots, m$)).
 Step 3: Calculate $c(c = \sum_{i \in N(i)} v_i / \sum_{i \in N(i)} g_i)$.
 Step 4: Compare C with C_j ($j \in M(j)$), mark the element satisfying the function $\text{Min}|c - C_j|$ as j , then the truck j is selected to load scattered goods.

Stage 2: One Truck Loading

To the truck j selected in the stage 1, loads the scattered goods using the one truck loading algorithm and marks the final loaded goods is $s_j^{(1)}(i)$.

Stage 3: Solution Optimization

The solution obtained in stage 2 is optimized in this stage. Namely, the calculation methods of c in stage 1 is adjusted from the $c = \sum_{i \in N(i)} v_i / \sum_{i \in N(i)} g_i$ to the

$c^{(2)} = \sum_{i \in \cup s_j} v_i / \sum_{i \in \cup s_j} g_i$. Marking the element satisfying the function $\text{Min} |c^{(2)} - C_j|$ as $j^{(2)}$, the truck $j^{(2)}$ is selected from trucks set $M(j)$ to load scattered goods, and the loaded goods set in the No. 2 iterative algorithm is $s_j^{(2)}(i)$.

The set of all loaded goods in the k^{th} iteration of the algorithm is $S^{(k)}(i) = \cup s_j^{(k)}(i)$. If the truck's utilization ratio in the two consecutive iterations get very close

$\left(\left| \sum_{i \in S^{(k+1)}} v_i / \sum_{j \in M^{(k+1)}(j)} V_j - \sum_{i \in S^{(k)}} v_i / \sum_{j \in M^{(k)}(j)} V_j \right| \leq \varepsilon, \varepsilon \text{ is the solution difference level} \right)$, the algorithm calculation is finished.

4 Numerical Investigation and Discussion

4.1 One Truck Loading

Comparing the C-W balance algorithm of one truck loading with other optimization methods, the concrete data of reference [10] is selected. The result is shown as follows.

Through simulating the computational algorithm to a large number of numerical investigations, some conclusions are drawn:

1) Comparing with the $A_k (k=0)$ algorithm, the C-W balance algorithm can obtain better solutions under the same computational complexity. Generally speaking, the truck's loading efficiency of the C-W balance algorithm, comparing with the $A_k (k=0)$ algorithm, can be improved by 5% to 20%.

2) Comparing with the $A_k (k=1)$ algorithm, the C-W balance algorithm can obtain satisfactory solutions in equal efficiency. But the computational complexity of $A_k (k=1)$ algorithm is $O(kn^{k+1})$, while the computational complexity of C-W balance algorithm is just $O(n \log n)$. With the increasing of scattered goods, the C-W balance algorithm runs fast.

4.2 $M (M > 1)$ Truck Loading

Through numerical simulation, the C-W balance algorithm is compared with the First-Fit algorithm. The relevant data of simulation are as follows: a) trucks: the cubage and

Table 1. One truck loading algorithm

Algorithm	$A_k (k = 0)$	$A_k (k \geq 1)$	C-W balance algorithm
Input	Goods set: $N(i) = \{1, 2, 3, 4, 5, 6, 7, 8\}$		
	Goods volume set: $N(v_i) = \{110, 108, 96, 80, 49, 50, 40, 7\}$		
	Goods weight set: $N(g_i) = \{64, 52, 50, 41, 22, 20, 14, 2\}$		
	Truck: cubage $V = 250$, carry capacity $G = 110$		
Loaded goods set	$p = \{1, 6, 7, 8\}$	$P = \{2, 6, 7, 5\}$	$P = \{2, 6, 7, 5\}$
Total loaded cubage	207	247	247
Cubage utilization rate	90%	98%	98%
Total loaded weight	100	108	108
Weight utilization rate	82%	98%	98%
Computational complexity	$O(n \log n)$	$O(kn^{k+1}) (k \geq 1)$	$O(n \log n)$

the weight are generated at random from the array (200-400), (100-200) respectively, and the quantity of trucks is 100; b) scattered goods: the volume and the weight are generated at random from the array (20-60), (10-40) respectively, and the quantity of all scattered goods is 5000. The simulation results are shown in Fig.1 and Fig.2.

The following observation can be seen in Fig.1 and Fig.2.

(1) For truck No. 75, 52, 88, 57, 62, 48, 16, 84, 39, 30, 96, the cubage utilization ratio (CUR) of First-Fit algorithm is slightly higher than that of the C-W balanced algorithm, and the First-Fit utilization is up to 8% higher than C-W balanced algorithm. For other trucks, the CUR of C-W balanced algorithm is higher. Especially in the later stage of problem solving, the disparity is more pronounced. Among these trucks, the CUR of truck No. 61, 13, 81, 77, 32, 46, and 87 is more than 20% higher. And to truck No. 61, the utilization is 28% higher using C-W balanced algorithm.

(2) For truck No. 11, 39, 30 the weight utilization ratio (WUR) of First-Fit algorithm is slightly higher than that of the C-W balanced algorithm, and the largest difference (truck No.11) is nearly 1.3%. For other trucks, the WUR of C-W balanced algorithm is higher, and this disparity is more obvious in the later stages of problem solving. For truck No. 99, 54, 13, 77, 32, 91, 46, 1, 70, 78, 24, 33, 92, 56, the WUR of C-W balanced algorithm is 15%. For truck No. 87, the disparity is as much as 21%.

(3) The numerical simulation results show the stability of C-W balance algorithm is better than that of the First-Fit algorithm. As shown in the trend-lines, the CUR curve and the WUR curve of C-W balance algorithm are relatively steady, and the fluctuation of the First-Fit algorithm is relatively large. Especially to the later stage of problem solving, the efficiency of the First-Fit algorithm drops significantly below that of C-W balance algorithm.

Fig. 1. CUR of M truck loading algorithm

Fig. 2. WUR of M truck loading algorithm

5 Conclusion and Application

In practice, because there are a great deal of loaded goods and loading trucks and calculating by hand is impossible, we can take some measures as follows: 1) using computer. Using the data of loaded goods and loading trucks as input, we run the computer program and output the loading plan according to the loaded goods. 2) When the condition is poor and no computer can be available, we can assemble the goods with maximum C-W rate and the goods with minimum C-W rate together among all loaded goods until no goods can be selected, then make a loading plan through calculating by hand. Because the optimization plan cannot be gained in each loading problem, so seeking a satisfying solution can simplify the problem and save the calculation time. Sometimes, it is necessary to simplify the loading restriction to speed up the loading and also can improve the integrative effect. The simplest way of the loading problem maybe as following: loading the big C-W rate goods and the small one together, laying the big C-W rate goods at the bottom of the truck and pile up with those small C-W rate goods, and loading the rest goods directly until the truck is filled. But at this time, these unloaded earlier should be loaded at the truck's border.

Acknowledgments

This work was supported by research grants from the National Natural Science Foundation of China (No.70332001, 70672040), National High Technology R&D Project of China (No. 2006AA04Z153) and HUST Doctor Foundation (No. D0540).

References

1. Dowsland, K.A., Dowsland, W.B.: Packing Problems. *European Journal of Operational Research*, 56(1992) 2-14
2. Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: Approximation algorithms for bin packing-an updated survey, in: Hochbaum, D.S. (Ed.), *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, Boston (1997)
3. Chen, B., Srivastava, B.: An improved lower bound for the bin-packing problem. *Discrete Applied Mathematics*, 66 (1996) 81-94
4. Stephen, M.H., Liu, Y.-H.: The application of integer linear programming to the implementation of a graphical user interface: a new rectangular packing problem. *Applied Mathematical Modeling*, 4(1995) 244-254
5. Berghammer, R., Reuter, F.: A linear approximation algorithm for bin packing with absolute approximation factor. *Science of Computer Programming*, 1(2003) 67-80
6. Scheithauer, G.: LP-based bounds for the container and multi-container loading problem. *International Transactions in Operational Research*, 6(1999) 199-213
7. Armstrong, R.D., Jin, Z.-Y.: Strongly polynomial simplex algorithm for bipartite vertex packing. *Discrete Applied Mathematics*, 2(1996) 97-103
8. Woeginger, G.J.: There is no asymptotic PTAS for two-dimensional vector packing. *Information Process Letter*, 64 (1997) 293-297
9. Chan, T.M.: Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 2(2003) 178-189
10. Sun, Y., Li, Z.-Z.: The Polynomial Algorithm for the Allocation Problem with Two Aims (In Chinese). *Changsha Railway University Transaction*, 2(1997) 33-39