European Journal of Operational Research 241 (2015) 39–49

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A goal-driven prototype column generation strategy for the multiple container loading cost minimization problem

Lijun Wei [a], Wenbin Zhu [b,*], Andrew Lim [c,d,1]

[a] School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, Jiangxi 330013, People's Republic of China
[b] School of Business Administration, South China University of Technology, Guangzhou 510640, People's Republic of China
[c] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371, Singapore
[d] Nanjing Univ, Sch Management & Engn, Int Ctr Management Sci & Engn, Nanjing 210008, Jiangsu, People's Republic of China

A R T I C L E I N F O

Article history:
Received 8 May 2013
Accepted 10 August 2014
Available online 15 August 2014

Keywords:
Multiple container loading cost minimization problem
Prototype column generation
Integer programming
Goal-driven search

A B S T R A C T

In the multiple container loading cost minimization problem (MCLCMP), rectangular boxes of various dimensions are loaded into rectangular containers of various sizes so as to minimize the total shipping cost. The MCLCMP can be naturally modeled as a set cover problem. We generalize the set cover formulation by introducing a new parameter to model the gross volume utilization of containers in a solution. The state-of-the-art algorithm tackles the MCLCMP using the prototype column generation (PCG) technique. PCG is an effective technique for speeding up the column generation technique for extremely hard optimization problems where their corresponding pricing subproblems are NP-hard. We propose a new approach to the MCLCMP that combines the PCG technique with a goal-driven search. Our goal-driven prototype column generation (GD-PCG) algorithm improves the original PCG approach in three respects. Computational experiments suggest that all three enhancements are effective. Our GD-PCG algorithm produces significantly better solutions for the 350 existing benchmark instances than all other approaches in the literature using less computation time. We also generate two new set instances based on industrial data and the classical single container loading instances.

1. Introduction

The efficient loading of items into containers is a fundamental problem in the shipping and logistics industries that arises whenever goods must be packed and transported. In many practical scenarios, choices of containers of different sizes and costs available, and the task is to select a set of containers that can hold all the goods while minimizing the cost. This problem is known as the *multiple container loading cost minimization problem* (MCLCMP).

The MCLCMP is formally defined as follows. We have M types of containers with dimensions $L_t \times W_t \times H_t,\ t = 1, \ldots, M$. The cost of the t-th container is C_t, and there are m_t containers available for container type t. We are also given N types of boxes. The dimensions of the i-th box type are $l_i \times w_i \times h_i,\ i = 1, \ldots, N$, and there are n_i boxes of type i. The objective of the MCLCMP is to pack all boxes into a set of containers so that the cost of the used containers is minimized. We assume that the boxes can only be placed with sides parallel to the sides of the containers (commonly referred to as *orthogonal packing*), and no two boxes in the same container may overlap. In some applications, it is desirable that all boxes are fully supported from below for stability. This arrangement is called the *full support constraint* in literature. Our approach works with or without the full support constraint.

This paper primarily considers a special case in which there are an unlimited number of available containers for each type (i.e., $m_t = \infty\ \forall t$). This is because the number of containers available is usually more than sufficient to contain all items to be loaded in practice, except in rare cases during peak periods. However, our technique can also be extended easily to handle a limited number of containers.

We build upon the prototype column generation (PCG) method proposed by Zhu, Huang, and Lim (2012) for the MCLCMP using a

* Corresponding author. Tel.: +86 13922106604.
E-mail addresses: villagerwei@gmail.com (L. Wei), i@zhuwb.com (W. Zhu), alim.china@gmail.com (A. Lim).
[1] Andrew Lim is currently on no pay leave from CityU of Hong Kong.

http://dx.doi.org/10.1016/j.ejor.2014.08.015
0377-2217/

goal-driven strategy. Our goal-driven prototype column generation (GD-PCG) algorithm improves on PCG in three respects. First, the PCG algorithm carries out the search in two stages and each stage approaches good solutions to the MCLCMP in its own dimension, whereas GD-PCG searches in both dimensions at the same time. Second, we extend the set cover formulation in Section 4 by introducing a new parameter γ, that reflects the estimated overall volume utilization of the containers in an MCLCMP solution. Third, once a solution is found, a goal-driven search described in Section 5.3 is applied in the neighborhood of the solution to improve it. Computational experiments reported in Section 6.1 suggest that all three enhancements are effective.

We compare our GD-PCG approach with existing MCLCMP approaches in the literature on the 350 benchmark instances proposed by Che, Huang, Lim, and Zhu (2011) in Section 6.2. The results show that GD-PCG outperforms all existing approaches in terms of solution quality and average computation time. We also generate two new set of instances based on real data from an international manufacturer of audio equipment and the classical single container loading instances.

2. Literature review

Under the improved typology for cutting and packing problems introduced by Wäscher, Haußner, and Schumann (2007), the MCLCMP can be considered as a variant of either the multiple stock-size cutting stock problem (MSSCSP) or the multiple bin-size bin packing problem (MBSBPP), depending on the heterogeneity of the boxes, where the objective is to minimize the cost of the containers.

Eley (2003) proposed a bottleneck assignment approach to the MCLCMP and some possible variants. The author first generated packing patterns using a tree search based heuristic, then used a set cover formulation for linear integer programming using pre-generated packing patterns. Che et al. (2011) adapted this set cover formulation by adding a loading factor parameter to exploit the excess capacity of the chosen containers, used three fast heuristic strategies to generate packing patterns, and performed a binary search on the loading factor. Zhu, Huang, et al. (2012) presented a PCG strategy for this problem to speed up the process of column generation. In solving the pricing problem during column generation, the authors used prototypes that approximated feasible solutions to the pricing problem rather than actual columns.

A special case of the MCLCMP is the 3-D bin packing problem (3D-BPP), in which there is only one type of container. The 3D-BPP is a reasonably well-studied problem (Alvarez-Valdes, Parreño, & Tamarit, 2013; Crainic, Perboli, & Tadei, 2008, 2009; Faroe, Pisinger, & Zachariasen, 2003; Fekete & van der Veen, 2007; Lodi, Martello, & Vigo, 2002; Martello, Pisinger, & Vigo, 2000; Parreño, Alvarez-Valdes, Oliveira, & Tamarit, 2008a; Verweij, 1996; Zhu, Zhang, Oon, & Lim, 2012). However, 3D-BPP approaches generally assume that the items cannot be rotated, which is an unrealistic assumption since for most practical applications some items can at least be rotated by 90 degrees. Furthermore, the fact that containers come in multiple standard sizes which creates a trade-off between the size of the container and its cost, is not reflected in the 3D-BPP.

One of the subproblems of the MCLCMP is the single container loading problem (SCLP), which is also well-studied in the literature. The current best approaches for the SCLP include those of Eley (2002), Gehring and Bortfeldt (2002), Bortfeldt, Gehring, and Mack (2003), Mack, Bortfeldt, and Gehring (2004), Lim and Zhang (2005), Moura and Oliveira (2005), Parreño, Alvarez-Valdes, Tamarit, and Oliveira (2008b), Fanslau and Bortfeldt (2010), Parreño, Alvarez-Valdes, Oliveira, and Tamarit (2010), Zhu, Qin, Lim, and Zhang (2012), Zhu, Oon, Lim, and Weng (2012), Wang, Lim, and Zhu (2013), and Lim, Ma, Qiu, and Zhu (2013). The earlier literature often recommended ways to adapt procedures for the SCLP for multiple containers. Possible strategies include the sequential strategy, which fills single containers in turn using SCLP approaches (Ivancic, Mathur, & Mohanty, 1989; Eley, 2002; Lim & Zhang, 2005); the pre-assignment strategy, where boxes are assigned to containers before loading (Terno, Scheithauer, Sommerweiß, & Riehme, 2000); and the simultaneous strategy, where multiple containers are considered in the loading of boxes (Eley, 2002).

3. Preliminaries

3.1. Set cover formulation for the MCLCMP

A solution to the MCLCMP can be defined by a list of packing patterns, where each pattern represents the packing configuration for one container. Let P denotes a set of candidate packing patterns, indexed by $j = 1, \ldots, |P|$. Each packing pattern j is represented by a column vector $\mathbf{A}^j = (a_{1j}, \ldots, a_{Nj})^T$, where a_{ij} is the number of boxes of type i used in the packing pattern. Let $\mathbf{A}$ be the matrix formed by all the column vectors corresponding to the packing patterns in P. If the container used by packing pattern j is of type t, then the cost associated with this pattern is $c_j = C_t$. Let the vector $\mathbf{x}$ be the integer decision variables, where x_j denotes the number of times the j-th packing pattern is used. A feasible solution to the MCLCMP can be obtained by solving the following model:

$$\mathbf{SCP}(P): \quad \text{Minimize} \quad z = \sum_{j=1}^{|P|} c_j x_j \tag{1}$$

$$\text{Subject to} \quad \sum_{j=1}^{|P|} a_{ij} x_j \geqslant n_i, \quad i = 1, \ldots, N \tag{2}$$

$$x_j \geqslant 0 \text{ and integer}, \quad j = 1, \ldots, |P| \tag{3}$$

where the objective function (1) seeks to minimize the total cost of all selected containers, and the inequalities (2) make sure that the selected patterns have packed enough boxes of each type where n_i is the number of boxes of type i to be packed. This set cover formulation was first proposed by Eley (2003) and later employed by Che et al. (2011). The set cover formulation for both 1D and 2D bin packing problem has also been proposed by Monaci and Toth (2006).

Note that if the full-support constraint is not imposed and $\mathbf{A}$ contains all possible feasible single container packing patterns, then we can solve the above integer program to find an optimal solution to the MCLCMP even though constraints (2) are inequalities rather than equalities. This is because if the selected packing patterns in the solution involve excess boxes of type i, then we can remove type i boxes from this solution until only the required number of boxes remains. The resulting packing patterns are still feasible, because the support of the boxes is not required.

3.2. Prototypes column generation

One difficulty in using the above set cover formulation to solve the MCLCMP is constructing the matrix $\mathbf{A}$ such that each column corresponds to a feasible packing pattern. A feasible packing pattern is a solution to the single container loading problem (SCLP), which is NP-hard, so finding these patterns can be very time consuming.