European Journal of Operational Research 214 (2011) 501–511

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

The multiple container loading cost minimization problem

Chan Hou Che [a], Weili Huang [a], Andrew Lim [a], Wenbin Zhu [b,c,*]

[a] Department of Management Sciences, City University of Hong Kong, Tat Chee Ave, Kowloon Tong, Hong Kong
[b] Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[c] Department of Logistics and Maritime Studies, Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 31 March 2010
Accepted 18 April 2011
Available online 27 April 2011

Keywords:
Packing
Heuristics
Container loading
Integer programming
Design of experiments

ABSTRACT

In the shipping and transportation industry, there are several types of standard containers with different dimensions and different associated costs. In this paper, we examine the multiple container loading cost minimization problem (MCLCMP), where the objective is to load products of various types into containers of various sizes so as to minimize the total cost. We transform the MCLCMP into an extended set cover problem that is formulated using linear integer programming and solve it with a heuristic to generate columns. Experiments on standard bin-packing instances show our approach is superior to prior approaches. Additionally, since the optimal solutions for existing test data is unknown, we propose a technique to generate test data with known optimal solutions for MCLCMP.

1. Introduction

Our team was contracted by a buying agent for a large multinational retailer to investigate better ways to formulate packing plans for the loading of goods into multiple containers. The agent's procurement department is responsible for ordering products from manufacturers located in Asia, inspecting the products to ensure quality, and finally shipping the products to retail stores spread across Europe. As a result, one of the procurement department's tasks is to arrange the products for shipping. Typically, the goods are loaded into containers of various standard sizes with different costs. The task is to select a set of containers that can contain all items while minimizing the total shipping cost.

This problem is a variant of a class of problems known as the multiple container loading problem (MCLP), also called the multiple container packing problem. The MCLP describes the scenario where, given a set of 3-D rectangular boxes of different types and a set of 3-D rectangular containers of different types, the objective is to store the boxes into the containers as efficiently as possible. There are many variants to the MCLP; the particular variant our paper addresses is referred to as the multiple container loading cost minimization problem (MCLCMP).

The MCLCMP is defined as follows. We are given a number of rectangular containers of M types, represented by $C_1, C_2, \ldots, C_M$, that differ in terms of dimensions and cost. The cost for each type is given by $c_1, c_2, \ldots, c_M$, and there are an unlimited number of each container available. We are also given a number of rectangular boxes of N types with different dimensions, represented by $B_1, B_2, \ldots, B_N$; there are n_j, $1 \leqslant j \leqslant N$ available boxes of type j. The objective of the MCLCMP is to produce a set of packing plans such that all boxes are orthogonally packed into the containers and the total cost of containers used is minimized. Our technique can also handle a limited number of containers (i.e., there are m_t, $1 \leqslant t \leqslant M$ available containers of type t) with minor modifications (as detailed in Section 8). We do not consider supporting area constraints, although our approach can be modified to do so.

Our approach is an extension of the integer linear programming formulation proposed by Eley (2003) for the MCLP. We add a new parameter to this formulation that controls the percentage of boxes to be packed, and perform a binary search on this variable. In addition, we devise three different strategies to quickly generate packing patterns for single containers along with a subroutine that augments the set of packing patterns while searching for solutions for the IP-problem. We tested this technique using the standard set of 47 bin-packing test instances proposed by Ivancic et al. (1989), and found that it produced superior results in comparison with other existing approaches.

However, the optimal solutions for these test instances are unknown, and so we cannot evaluate the objective performance of our approach based on this test set. Hence, we also devise a technique for generating test instances for the MCLCMP with verifiably optimal known solutions.

The remainder of this paper is organized as follows. We first provide an overview of the existing literature related to our study

* Corresponding author at: Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Tel.: +852 64067667.

E-mail address: i@zhuwb.com (W. Zhu).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.04.017

in Section 2. Our new parameterized linear integer programming formulation is presented in Section 3; we perform binary search on the parameter based on a number of generated packing patterns. Section 4 describes our three strategies for generating packing patterns, which are used by the binary search algorithm given in Section 5. In order to obtain a more objective evaluation of the strength of our approach, we describe a new technique in Section 6 for generating test instances for this problem with known optimal solutions. We test our approach on both the bin-packing instances used in current research and on our new instances, and present the results in Section 7. Our paper concludes in Section 8, where we suggest ways to further extend our technique to handle other variants of the MCLP.

2. Related work

Following the improved typography for cutting and packing problems introduced by Wäscher et al. (2007), the MCLCMP can be considered 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.

There is little existing literature that examines the MCLP and its variants directly. Takahara and Miyamoto (27) studied a similar problem where the objective was to minimize wasted volume, which is equivalent to minimizing the total capacity of containers used. This may be regarded as a variant of the MCLCMP where the containers of various types have equal unit costs. An evolutionary approach was proposed, where a pair of sequences (one for boxes and one for containers) is used as the genotype, and a heuristic was used to determine the loading plan given the sequence of boxes and containers. The MCLCMP and some possible variants were also studied by Eley (2003) who proposed a bottleneck assignment approach. The author used a set cover formulation for linear integer programming using pre-generated packing patterns, which were found using a tree search based heuristic. We adapted this set cover formulation in our work; details are given in Section 3. Set-cover-based heuristic has also been proposed by Monaci and Toth (2006) for both 1D and 2D bin packing problems.

Although the 3-D bin packing problem (3D-BPP) is reasonably well-studied (Verweij, 1996; Lodi et al., 2002; Crainic et al., 2008; Fekete and van der Veen, 2007; Crainic et al., 2009; Faroe et al., 2003; Parreño et al., 2008; Martello et al., 2000), this problem differs from multiple container loading in two crucial ways. Firstly, 3D-BPP approaches generally assume that all items cannot be rotated; this is an unrealistic assumption since most items can at least be rotated by 90°. Secondly, 3D-BPP assumes that all bins are of the same size. However, containers come in multiple standard sizes, and there is a trade-off between the size of the container and its cost. Both of these factors must be addressed for the MCLCMP.

Earlier literature often recommended ways to adapt procedures for the well-analyzed single container loading problem (SCLP) for multiple containers. Possible strategies include the sequential strategy, where single containers are filled in turn using SCLP approaches (Ivancic et al., 1989; Bischoff and Ratcliff, 1995a,b; Eley, 2003; Lim and Zhang, 2005); the pre-assignment strategy, which first assigns boxes to containers before loading (Terno et al., 2000); and the simultaneous strategy, which considers multiple containers during the loading of boxes (Eley, 2002).

Our technique makes use of SCLP heuristics as sub-routines. Dyckhoff and Finke (1992) provided an excellent review of the SCLP, and Christensen and Rousøe (2009) also provided a brief introduction of some approaches in their technical report.

3. Linear integer programming formulation for the MCLCMP

Our approach is an extension of the set partitioning formulation proposed by Eley (2003) for the multiple container loading problem. However, the term *set partitioning formulation* used in the original publication is a misnomer since the formulation is in fact a set *cover*; we will henceforth refer to the formulation as a set cover formulation. We assume that there is a set P of single container packing patterns indexed by i, and the box types are indexed by j. Each packing pattern p_i fills a container with associated cost c_i with b_{ij} boxes of type j. Let x_i be the integer decision variable for the ith column that represents the number of packing patterns p_i used. A feasible solution to MCLCMP can be obtained by solving the following optimization model:

$$SC: z = \min \sum_{p_i \in P} c_i x_i \qquad (1)$$

$$\text{s.t.} \sum_{p_i \in P} x_i b_{ij} \geqslant n_j, \quad \forall j \qquad (2)$$

$$x_i \geqslant 0 \quad \text{and integer}, \quad i = 1, \ldots, |P|, \qquad (3)$$

where the objective function (1) computes the total cost of all selected packing patterns, and the inequality (2) ensures that the selected packing patterns have sufficient space to pack the boxes of each type, where n_j is the number of boxes to be packed for box type j.

It is clear that an optimal solution of the basic model SC is a feasible solution for the MCLCMP instance. However the solution is often suboptimal due to two reasons. Firstly, the set of packing patterns P must encompass the set of all valid packing patterns, and this is computationally prohibitive except for very small problem instances. As a result, a sample of valid packing patterns must be taken.

Secondly, the selected containers may offer excessive capacity for some box types, i.e., for some j the inequality (2) is strictly greater. The presence of excessive capacity offers an opportunity to reduce the cost by selecting a set of containers with smaller capacity. Hence, we introduced a loading factor a to inequality (2), and parameterized the model as follows:

$$SC(a): z = \min \sum_{p_i \in P} c_i x_i, \qquad (4)$$

$$\text{s.t.} \sum_{p_i \in P} x_i b_{ij} \geqslant a \cdot n_j, \quad \forall j, \qquad (5)$$

$$x_i \geqslant 0 \quad \text{and integer}, \quad i = 1, \ldots, |P|. \qquad (6)$$

The loading factor a is the percentage of boxes that are guaranteed to be loaded into selected containers. For example, the solution found by SC(0.97) will guarantee that at least 97% of each type of box can be loaded into the containers selected, while the remaining 3% has no guarantee. It may be possible to load the unguaranteed 3% into the containers selected by SC(0.97), whereupon we would obtain a feasible solution of MCLCMP. Note that the cost of the optimal solution of SC(a) is a non-decreasing function of a because if $a < b$, then any feasible solution to SC(b) is also a feasible solution to SC(a) (i.e., an optimal solution of SC(a) is at least as good as an optimal solution for SC(b)).

We make use of this parameterized formulation as follows. First, we generate a set P of packing patterns, which are solutions to the SCLP, using three strategies that employ fast heuristics for the SCLP. We then perform binary search on the range $0.01 \leqslant a \leqslant 1.0$ for *max_iter* iterations, employing a commercial IP solver for each iteration. If there are boxes that are not loaded in a given iteration, we invoke a routine that attempts to load these boxes without increasing the number of containers. This routine has the side-effect of possibly increasing the number of packing patterns (i.e., increasing the size of P).