Computers & Industrial Engineering 108 (2017) 149–164

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Heuristic algorithm for the container loading problem with multiple constraints

Liu Sheng [a,*], Shang Xiuqin [a], Cheng Changjian [b], Zhao Hongxia [a], Shen Dayong [c], Wang Feiyue [a]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, 100190 Beijing, China
[b] Institute of Process Engineering, Chinese Academy of Sciences, 100190 Beijing, China
[c] Qingdao Academy of Intelligent Industries, 266000 Qingdao, Shandong, China

ARTICLE INFO

Article history:
Received 9 January 2016
Received in revised form 6 March 2017
Accepted 13 April 2017
Available online 21 April 2017

Keywords:
Packing
Container loading
Complete shipment constraint
Shipment priority
Simulated annealing

ABSTRACT

This paper addresses the container loading problem with multiple constraints that occur at many manufacturing sites, such as furniture factories, appliances factories, and kitchenware factories. These factories receive daily orders with expiration dates, and each order consists of one or more items. On a particular day, certain orders expire, and the expiring orders must be handled (shipped) prior to the non-expiring ones. All of the items in an order must be placed in one container, and the volume of the container should be maximally utilized. A heuristic algorithm is proposed to standardize the packing of (order) items into a container. The algorithm chooses the expiring orders first before handling the non-expiring orders. In both steps, the algorithm first selects a collection of orders by considering a simulated annealing strategy and subsequently packs the collection of orders into the container via a tree-graph search procedure. The validity of the algorithm is examined through experimental results using BR instances.

1. Introduction

Many factories, such as furniture factories, appliance factories, and kitchenware factories, are confronted with the problem of packing item orders. Every day, these factories receive orders from buyers worldwide and ship the items to the buyers in containers. The factories want to load as many items as possible into a container to reduce shipping costs. Each order consists of one or more different items. On any given day, certain orders expire; the expiring orders must be packed and shipped prior to the non-expiring ones. If one item of an order is packed in the container, all items of the order must be packed in the same container. If one item of an order is not packed in the container, all items of the order must not be packed in the container. To protect the items, each item must be loaded with its height parallel to the height of the container. The bottom of each item must be supported by the container floor or by the top of a single item to simplify the unloading process and to ensure the stability of the items.

According to the typology proposed by Dyckhoff (1990), this issue is a 3/B/O problem. According to the recent typology proposed by Wäscher, Haußner, and Schumann (2007), this issue is a three-dimensional single knapsack problem (**3D-SKP**) or a three-dimensional single large object placement problem (3D-SLOPP). Following the review of the paper by Bortfeldt and Wäscher (2013), **3D-SKP** in this paper is a three-dimensional, container-loading problem (**3D-CLP**) with the orientation constraint (**C1**), the stability constraint (**C2**), the guillotine-cutting constraint (**C3**), the complete-shipment constraint (**C4**), and the loading priority constraint (**C5**).

The rest of the paper is organized as follows: In Section 2, we provide a literature review of three-dimensional container-loading problems. In Section 3, we review the methods that solve the container loading problem of item orders. In Section 4, we analyse the results of the presented method based on simulations created from BR instances (Bischoff & Ratcliff, 1995; Davies & Bischoff, 1999). Finally, in Section 5, we present the paper's conclusions perspectives for future research.

2. Literature review

Since the seminal work by George and Robinson (1980), **3D-CLP** has received increasing attention from academic researchers and

* Corresponding author.
E-mail addresses: liusheng7801@163.com, sheng.liu@ia.ac.cn (L. Sheng), xiuqin.zhao@ia.ac.cn (S. Xiuqin), changjian.cheng@ia.ac.cn (C. Changjian), hongxia.zhao@ia.ac.cn (Z. Hongxia), dayong.shen@qaii.ac.cn (S. Dayong), feiyuewang@gmail.com (W. Feiyue).

http://dx.doi.org/10.1016/j.cie.2017.04.021
0360-8352/

industry professionals. Bischoff and Ratcliff (1995) provide an excellent overview of the practical requirements that may be imposed on this problem. Bortfeldt and Wäscher (2013) provide a comprehensive scheme to categorize the constraints of loading a container and determine that the existing approaches have limited practical value because they do not sufficiently address the constraints encountered in practice. In this paper, we discuss **3D-CLP** with several practical constraints (described in the Introduction section). To the best of our knowledge, there are no published approaches that address **3D-CLP** using the practical constraints we investigate. We will briefly classify and discuss several of the recent advances in this problem.

3D-CLP is considered an NP-hard (Bischoff & Marriott, 1990) problem. Exact algorithms are usually constrained by a situation called the combinatorial space explosion when the number of item types increases. Consequently, notably few exact algorithms exist.

Martello et al. (2000); also see Den Boef et al., 2005) present a branch-and-bound algorithm that addresses the single knapsack problem and—based on this algorithm—a branch-and-bound method for addressing the single bin-size, bin-packing problem. The orientation of all of the items are fixed, and no further constraints are considered.

Hifi (2004) introduces an exact depth-first search and a dynamic programming algorithm for solving the 3D SLOPP in a cutting context. The orientation and the guillotine cutting constraint are both considered. The number of items per type is unlimited.

Fekete, Schepers, and Van der Veen (2007) develop an exact algorithm for higher-dimensional orthogonal packing problems, where the small items have no fixed orientations, and no other constraints are considered.

Heuristic methods prove to be a more realistic alternative for addressing **3D-CLP**. Although heuristic methods may only find sub-optimal solutions, they can produce sufficiently good solutions in a reasonable timeframe.

Orientation and stability constraints are the most studied constraints in the literature. In **3D-CLP**, small items may have at most six orthogonal orientations in the container. Pisinger (2002) and Egeblad and Pisinger (2009) assume that the small items may be rotated in any orthogonal direction. Most heuristic methods (e.g., Bischoff, Janetz, & Ratcliff, 1995; Bortfeldt & Gehring, 2001; Fanslau & Bortfeldt, 2010; Gehring & Bortfeldt, 2002; He & Huang, 2010, 2011; Huang & He, 2009; Lim, Ma, Xu, & Zhang, 2012; Lim, Rodrigues, & Wang, 2003; Moura & Oliveira, 2005; Parreño, Alvarez-Valdés, Tamarit, & Oliveira, 2008; Zhu, Huang, & Lim, 2012; Zhu & Lim, 2012) assume that some orientations are forbidden.

Load stability is often considered the most important issue after container space utilization in the literature (e.g., Bischoff & Ratcliff, 1995; Bortfeldt, Gehring, & Mack, 2003; Eley, 2002; Fanslau & Bortfeldt, 2010; Gehring & Bortfeldt, 2002; Liu, Tan, Xu, & Liu, 2014; Ren, Tian, & Sawaragi, 2011; Zhang, Peng, & Leung, 2012; Zhu & Lim, 2012; Zhu et al., 2012). Fanslau and Bortfeldt (2010) and Zhu et al. (2012) consider two situations where small items are fully or partially supported. Partial support is required by Gehring and Bortfeldt (1997) and Mack, Bortfeldt, and Gehring (2004).

The guillotine cutting constraint is often viewed from a loading perspective. A guillotine pattern represents a type of loading pattern that can be packed easily. A loading pattern is said to be *guillotineable* if it can be obtained by a series of "cuts" parallel to the container faces (especially the vertical faces). The guillotine cutting constraint is considered in Hifi (2002), Morabito and Arenalest (1994), and Liu et al. (2014). The loading pattern obtained by Pisinger (2002) is guillotineable although it is not declared as such.

In practice, the available container space is not sufficiently large to accommodate all small items, and the loading of some items may be more desirable than the loading of others. Thus, shipment priorities (Bortfeldt & Wäscher, 2013, also called loading priorities by Bischoff & Ratcliff, 1995) exist for some items. The shipment priority constraint is considered by Bortfeldt and Gehring (1999), Ren et al. (2011), and Wang, Lim, and Zhu (2013).

Certain subsets of loaded items may include functional or administrative supplies (Bischoff & Ratcliff, 1995). If one item of a subset is loaded, all other items of that subset must also be loaded. If one item cannot be loaded, no item of the subset will be loaded at all. Two cases can be distinguished: In the first case, all items of a subset must be included in the shipment. In the second case, all items of a subset have to be loaded into the same container. Eley (2003) considers the first case in a multiple heterogeneous large object placement problem.

A heuristic algorithm for container-loading of furniture, by Egeblad, Garavelli, Lisi, and Pisinger (2010), is remarkable in that a large variety of irregular items are considered and many practical constraints are satisfied. However, the loading priority constraint and the complete-shipment constraint are not considered. Lim, Ma, Qiu, and Zhu (2013) consider the axle-weight constraint when solving the single container loading problem. Chen, Lee, and Shen (1995) provide an analytical model for the container loading problem. Junqueira, Morabito, and Sato Yamashita (2012) propose MIP-based approaches for the container loading problem with multi-drop constraints. Liu, Zhao, Dong, and Cheng (2016) present a heuristic algorithm for container loading of pallets with infill boxes.

The great majority of the methods mentioned above obey the orientation constraint and the stability constraint. Some methods also include additional constraints, e.g., such as the guillotine constraint, the loading priority constraint and the complete-shipment constraint. However, no approaches simultaneously consider all of the five constraints. Thus, we will develop a new algorithm that can handle the container loading problem with all of the five constraints.

3. Method for the container loading problem of item orders

3.1. Problem definition

The **3D-CLP** in this paper is called the container-loading problem with multiple constraints **CLPMC**. **CLPMC** is defined as follows.

A factory has m unfinished orders that are set to expire and n unfinished orders that do not expire. The total number of items in the $m+n$ orders is k. The k items are characterized by lengths $(l_1, l_2, \ldots, l_k)$, widths $(w_1, w_2, \ldots, w_k)$, and heights $(h_1, h_2, \ldots, h_k)$. The container C is characterized by the length L, the width W and the height H. The objective is to load a subset of these $m+n$ orders with maximum item volume into C. Additionally, the five constraints (**C1**-**C5**) must be fulfilled.

We define:

$$o_{ij} = \begin{cases} 1; & \text{if the } j\text{th item is in the } i\text{th expiring order}, \\ 0; & \text{otherwise}. \end{cases} \quad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, k), \tag{1}$$

$$r_{ij} = \begin{cases} 1; & \text{if the } j\text{th item is in the } i\text{th unexpiring order}, \\ 0; & \text{otherwise}. \end{cases} \quad (i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, k), \tag{2}$$

$$a_i = \begin{cases} 1; & \text{if the items of the } i\text{th expiring order is packed in the container}, \\ 0; & \text{otherwise}. \end{cases} \quad (i = 1, 2, \ldots, m), \tag{3}$$