Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-3 (2015) 490–495

A layer-building algorithm for the three-dimensional multiple bin packing problem: a case study in an automotive company

Rommel D. Saraiva, Napoleão Nepomuceno, Plácido R. Pinheiro

Graduate Program in Applied Informatics, University of Fortaleza
Av. Washington Soares, 1321 - J30, 60.811-905, Fortaleza - Brazil
(e-mails: rommelds@edu.unifor.br, {napoleaovn,placido}@unifor.br)

Abstract: The three-dimensional multiple bin packing problem (3D-MBPP) consists of packing a set of items into a number of bins with different dimensions so as to optimize a given objective function, e.g., minimize the number of bins used to pack the items. In this paper, we consider a real world 3D-MBPP with several cargo constraints that arises from an automotive maker. We propose an algorithm that first builds horizontal layers of identical items and then, according to different selection criteria, greedily generates packing patterns by loading one layer at a time. Computational experiments performed on benchmark instances are reported, and the results are compared to those achieved through a well-known constructive heuristic.

Keywords: Combinatorial optimization, Cutting and packing, Three-dimensional bin packing, Container loading.

1. INTRODUCTION

Renault S.A.[1] is a globally operating French manufacturer which focuses on the production of cars and vans. The company has presence in more than 120 countries, and the transportation of vehicle parts and accessories worldwide to serve its markets has become a critical activity. Recently, the EURO Special Interest Group on Cutting and Packing (ESICUP)[2] has launched a challenge[3], sponsored by Renault, that addresses a three-dimensional multiple bin packing problem (3D-MBPP) with several specific cargo constraints. The problem can be classified as a multiple bin-size bin packing problem within the typology by Wäscher et al. (2007).

Basically, the multiple bin-size bin packing problem can be summarized as follows: Consider a set of three-dimensional rectangular items (or boxes) grouped into m types. For each item type $i = 1, \ldots, m$, characterized by its length l_i, width w_i, and height h_i, there is a given demand d_i associated with it. Consider, as well, a set of three-dimensional rectangular bins (or containers) grouped into n types, each one with a specific length L_j, width W_j, and height H_j, $j = 1, \ldots, n$. The problem consists of orthogonally packing the set of items into a number of bins (without overlapping or protrusion) such that a predetermined objective function is optimized, e.g., the total number of bins used is minimized.

Bin packing problems are strongly NP-hard (Martello et al., 2000) and extremely difficult to solve in practice. Hence, heuristics are often used as alternative methods to exact algorithms for obtaining feasible solutions within an acceptable execution time, even if they cannot guarantee to find an optimal solution.

In the literature, the few papers that treat the 3D-MBPP under a heuristic perspective are motivated by real cases observed in industries, such as a biscuit factory (Brunetta and Grégoire, 2005), an automobile manufacturer in Turkey (Ertek and Kilic, 2006) and a logistic center (Alvarez-Valdés et al., 2013). In this work, we propose a constructive layer-building (CLB) algorithm to tackle the Renault/ESICUP challenge. Computational experiments performed on benchmark data sets are reported and discussed here. Furthermore, as a mean to comparatively assess the quality of our method, we have also implemented a well-known wall-building heuristic (George and Robinson, 1980).

The remainder of the paper is organized as follows. In Section 2, we introduce the Renault/ESICUP challenge more in details. Section 3 is devoted to the description of our constructive layer-building algorithm. In Section 4, we evaluate the performance of our method on benchmark problem instances. Some final remarks and comments on future work conclude the paper with Section 5.

2. THE RENAULT/ESICUP CHALLENGE

The Renault/ESICUP challenge addresses a 3D-MBPP for which, in addition to classical geometric constraints, many practical cargo issues are considered. In the sequel, we present notations, optimization criteria, and constraints of the problem under investigation.

2.1 Items

Item types are modelled as a 5-tuple $(l_i, w_i, h_i, r_i, p_i)$, where l_i, w_i, and h_i are the three space dimensions (length,

[1] http://www.renault.fr
[2] http://www.fe.up.pt/esicup
[3] http://challenge-esicup-2015.org

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.06.129

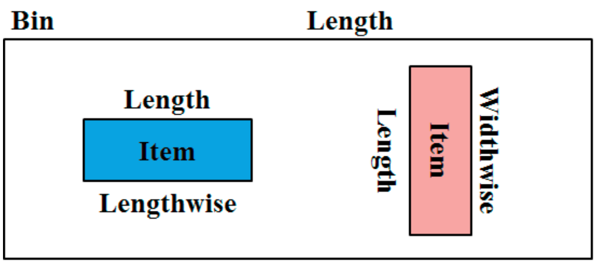

Fig. 1. Top view of two items with different orientations.

width, and height, respectively), r_i is its weight, and p_i represents the material of the item type (metal, cardboard, plastic, or wood). There is a demand d_i associated with each item type i. Whenever possible, items may only be rotated in the horizontal plane. If an item cannot be rotated, it will either be fixed lengthwise (with dimension l_i parallel to dimension L_j of the bin) or widthwise (with dimension w_i parallel to dimension L_j of the bin), as shown in Fig. 1.

2.2 Rows

A row is a sequence of contiguous items. Each row may be itself organized lengthwise or widthwise. If a row is organized lengthwise (resp. widthwise), then it is composed of contiguous items that have their origins in the same y-coordinate (resp. x-coordinate), as illustrated in Fig. 2. The number of items in a row is bounded by a problem parameter β^R.

The horizontal dimensions of a row are those represented by the rectangular envelope of its items. To prevent the existence of holes of significant magnitude, the sizes of the items in the dimension that is orthogonal to the row may differ from each other in at most a given ratio δ^R.

2.3 Layers

A layer is composed of contiguous rows and may be itself organized lengthwise or widthwise. The rows of a lengthwise (resp. widthwise) layer are packed with their origins in the same y-coordinate (resp. x-coordinate). The number of rows allowed for each layer is delimited by a problem parameter β^L. All items in a layer (and, therefore, in a row) must have the same height, except if there is no other layer on its top.

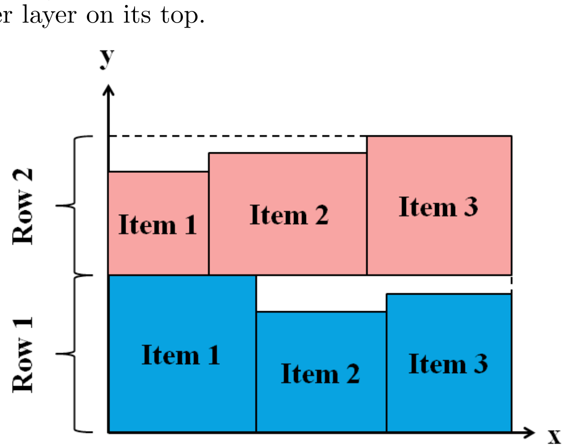

Fig. 2. Top view of two lengthwise rows.

Analogously, the horizontal dimensions of a layer are those of the rectangular envelope of its rows, and the sizes of the rows in the dimension that is orthogonal to the layer can only differ from each other in at most a given ratio δ^L.

2.4 Stacks

Contiguous layers can be packed along the height dimension to form a stack. The height of the stack is, therefore, the sum of the heights of its layers, and it must conform to the height of the bin. Layers in a stack have to be packed bottom-up-wise by decreasing weight. In addition, the layers of a same stack cannot differ, in both horizontal dimensions, by than a given ratio δ^S. The same constraint holds for the width dimensions. However, the top layer may be largely smaller than the other layers of the stack.

Metallic items have to be packed separated from non-metallic items in a stack. Particularly, metallic stacks are composed of single items packed one above the others. Conversely, non-metallic stacks require special considerations. First, let us define the length (resp. the width) of a stack as the largest length (resp. width) among all of its layers. The ground area of a stack is the rectangular envelope of the projections of the layers on the ground plan. By dividing the total weight of the items of the stack by its ground area, we obtain the density of the stack. For non-metallic stacks, the density cannot be larger than a problem parameter α. Finally, let the bottom layer of a stack be its lowest layer. Items in the bottom layer must support the weight of the stack. Again, for non-metallic stacks, the following constraint has to be satisfied:

$$\frac{area(i) \times weight(bottomlessstack)}{area(bottom)} \leq \mu \qquad (1)$$

where $area(bottom)$ is the total area of the bottom layer; $area(i)$ is the area of an item i in the bottom layer; $weight(bottomlessstack)$ is the weight of the stack minus the weight of the bottom layer; and μ is an input data.

2.5 Bins

There are several possible bin types, each one of infinite number. Bin types are modelled as a 4-tuple (L_j, W_j, H_j, R_j), where L_j, W_j ,and H_j denote, respectively, the length, the width, and the height of a bin of type j. The maximum weight that a bin of type j can hold is R_j (i.e., the sum of the weights of the items packed into the bin must be smaller than or equal to R_j). A bin can contain several stacks that are packed orthogonally on its floor with no overlap or protrusion.

There is a special bin (called bin 0) which stores items that are not sent in the current batch. For a given item type i, with demand equals to d_i, there cannot be more than $\lfloor d_i \times \gamma \rfloor$ items in this special bin, where γ is a problem parameter in the real interval $[0, 1]$. In the solution, the bin 0 must be that one with the smallest volume used, i.e., the sum of the volumes of the stacks including the volume lost inside them.