Lexicographic optimization for the multi-container loading problem with open dimensions for a shoe manufacturer

Manuel V.C. Vieira[1] and Margarida Carvalho[2]

[1]Departamento de Matemática, NOVA School of Science and Technology & CMA, Universidade NOVA de Lisboa, Portugal, email: mvcv@fct.unl.pt
[2]Département d'Informatique et de Recherche Opérationnelle & CIRRELT, Université de Montréal, Montréal, Québec H3T 1J4, Canada, email: carvalho@iro.umontreal.ca

August 16, 2023

Abstract

Motivated by a real-world application, we present a multi-container loading problem with 3-open dimensions. We formulate it as a biobjective mixed-integer nonlinear program with lexicographic objectives in order to reflect the decision maker's optimization priorities. The first objective is to minimize the number of containers, while the second objective is to minimize the volume of those containers. Besides showing the NP-hardness of this sequential optimization problem, we provide bounds for it which are used in the three proposed algorithms, as well as, on their evaluation when a certificate of optimality is not available. The first is an exact parametric-based approach to tackle the lexicographic optimization through the second objective of the problem. Nevertheless, given that the parametric programs correspond to large nonlinear mixed-integer optimizations, we present a heuristic that is entirely mathematical-programming based. The third algorithm enhances the solution quality of the heuristic. These algorithms are specifically tailored for the real-world application. The effectiveness and efficiency of the devised heuristics is demonstrated with numerical experiments.

1 Introduction

Problem statement Packing problems play an important role in the practical industrial processes. Typically, the goal of these problems is to find an optimized packing of a given set of items into a set of containers. Although packing problems have been widely studied in the literature, many questions remain open and unsolved. In this paper, we present a container loading problem which besides encompassing various of the typical challenges faced by packing problems, it requires a lexicographical optimization, i.e., a hierarchical ordering of two natural objective functions. Briefly, we aim to pack several small items amongst containers whose dimensions are open. This packing is guided by the minimization in the first objective of the number of used containers and then, in the second objective, by the minimization of their total volume subject to the number of containers decided in the first optimization.

Application The problem described is motivated by a real-world application suggested by a shoe manufacturer. The shoe manufacturer is specialized in children's footwear production. The children's shoe sizes range from 18 to 40. Each model size is packed in a shoe box. There is a limited number of available shoe boxes sizes, which are listed in Table 1. The footwear ordered by each customer is packed into one or several large cardboard boxes and these cardboard boxes are size customized for each set of shoe boxes to pack. The cardboard boxes will henceforth be referred to as containers. Each container can pack different shoe sizes and models, and it is to be delivered to one single customer. The goal is to pack the full order with the least number of containers and, as the containers are not fixed size, also minimizing their total volume. Each shoe box is packed with the label facing up for easy inspection and it can be rotated according with Figure 1. More details about this application can be read at the nontechnical magazine Impact, see Vieira and Flora (2020). Therefore, contrary to other industries, the customization of containers occurs and the consequent sequential optimization of two objective functions, results in a novel problem. We stress that the open-dimensions element is not an isolated case of this specific manufacturer as others have been described by Tsai and Li (2006), Tsai et al. (2015), and Junqueira and Morabito (2017).

Contributions and paper structure The paper is organized as follows. Section 2 reviews the related literature on both topics, lexicographic optimization and the container loading problem. Throughout Section 3, we introduce the lexicographic mixed-integer nonlinear mathematical formulation of the container loading problem with 3-open dimensions and we show that it is NP-hard. In Section 4, we reduce our problem to a series of parametric nonlinear mixed integer mathematical programs. Section 5 provides bounds to our problem which in Section 6 are used *(i)* to reduce the number of parametric problems to be solved by our exact approach and *(ii)* to simplify the nonlinearities of the model, leading to a non-exact algorithm. A variant of the non-exact method is proposed with the goal of improving the solution quality. Section 7 presents the computational results in three types of instances: the ones provided by the manufacturer, randomly generated instances, and a set of controlled instances for which we know the optimal value. These experiments allow us to validate the applicability of our methodologies, evaluate their scalability, as well as to analyze the quality of our heuristics. We remark that our methodology is described in terms of the case with 3-open dimensions. However, since the particular case of the shoe manufacturer does not allow to stack containers, our experiments only consider 2-open dimensions.

Since our work is based on a real-world application, it further justifies the practical interest on packing problems with open dimensions, expanding the literature in this direction. Moreover, it contributes with new algorithmic ideas to scale methodologies considering a multi-container setting. Finally, it combines in a new model, a sequential optimization problem which not only enlarges the applicability of lexicographic optimization formulations but also motivates the development of parametric frameworks.

2 Literature review

In many real-world applications the assumption that decisions are made by a single decision maker or guided by a single objective might be a relevant limitation. Next, we review the relevant works on multiobjective in the context of packing problems and we refer the reader to Ehrgott (2005) for a wide view on such problems.

Multiobjective optimization in packing problems Most work in packing problems consider either to minimize the number of containers or to minimize the volume of the container (or somehow equivalent, to maximize the volume utilization). Up to now, multiobjective research works combine just one of the previous objectives with other objectives, but not both. Minimizing the number of boxes and minimizing the total volume coincide in the continuous case, whereas they do not usually coincide when the items cannot be split.

Minimizing the number of containers is combined with the objective of minimizing the number of packing patterns in Liu et al. (2012). In the work of Liu et al. (2014), the number of packing patterns is also minimized but this time conflicting with the minimization of wasted space. The recent work of

Araya et al. (2020) and the previous work of Zheng et al. (2015) present a biobjective variant of the single container loading problem combining the objectives of maximizing the volume utilization and the total profit of the loaded items.

Dahmani et al. (2013) describe a bin-packing problem which minimizes the number of bins subject to the usual constraints and with a constraint fixing the height of the bins. Then, the height of the bins became a variable and that constraint is replaced by a second objective minimizing the height of the bins. Their problem does not decide the position of the items to be packed, instead, it just decides the number of bins, the assignment of items to bins, and the height of the bins, while seeking Pareto solutions. They also describe an application of assignment of jobs to machines. From the literature, this work is the most related with ours: on the one hand, our problem generalizes it by accounting in the decision variables for the items' positions and all 3 dimensions of the bins; on the other hand, we seek a specific Pareto solution satisfying the lexicographic order of two objectives.

In addition to the minimization of the number of bins, the works of Fernández et al. (2013) and Liu et al. (2008) also minimize the load imbalance. Khanafer et al. (2012) consider a biobjective problem where the second objective minimizes the number of conflicts. This assumes the existence of pairs of items to pack that we should avoid to place together in the same bin. Spencer et al. (2019) present a multiobjective bin packing problem with three competing goals for storing cooling objects. Besides minimizing the number of boxes, it minimizes the average initial heat of each box and the time until the boxes can be moved to the storefront.

Concerning lexicographic optimization, the work of Coniglio et al (2019) proposes an exact lexicographic dynamic pricing algorithm for solving the fractional bin-packing problem. There is also a discussion paper (Hessler et al. (2020)) which optimizes in a strictly lexicographic sense several objective functions: the total number of trucks, the number of refrigerated trucks, the number of refrigerated trucks which contain frozen products, the number of refrigerated trucks which also transport standard products, and then minimize the product splitting.

Related packing problems The multi-container loading problem with three open dimensions was first approached by Vieira et al. (2021). This paper describes an approach which performs three independent decision steps: 1) the number of containers that are previously fixed for each order; 2) the shoe boxes are sorted by size and they are assigned up to 20 boxes to each container, according to their position and, finally, 3) the volume of each container, the position, and the orientation of the boxes are decided using a mixed-integer linear program (MILP). On the contrary, in this paper, we introduce for the first time a global approach where the sequential optimization encloses in full these three steps. In other words, each decision will anticipate the ones that follow. Such anticipation allows for better integration of decisions and, in particular, it avoids infeasible solutions. For example, if the number of containers disregards limitations on their sizes, it can happen that the boxes do not fit in the limited amount of available containers.

Besides the previously mentioned paper, to the best of our knowledge, the multi-container loading problem where the containers have the 3 open dimensions was never studied before. Few works considering a single container with three open dimensions have been published. Tsai and Li (2006) and Tsai et al. (2015) propose a mixed-integer linear program where the objective function was the container's volume which was approximated by a piecewise linear function. Junqueira and Morabito (2017) also use the piecewise linearization technique and present grid-based position formulations to obtain an alternative MILP formulation.

Concerning the problem of loading several containers, Silva et al. (2019) present a comparative study of exact methods for the three-dimensional cutting and packing problems. Alonso et al. (2017) describe the multi-container loading problem where axle-weight constraints are also included. They minimize the number of trucks while balancing the mass center with the truck axles. Toffolo et al. (2017) also explore the three-dimensional multi-container loading problem. In all these papers, the container dimensions are fixed.

For a wide view of the packing problems, please refer to the review papers of Bortfeldt and Wäscher (2013), Zhao et al. (2016), and Delorme et al. (2016). While Bortfeldt and Wäscher (2013) review constraints for the loading problem, Zhao et al. (2016) compare algorithms for the 3D container loading problem, and Delorme et al. (2016) surveys bin-packing and related problems.

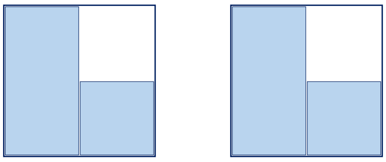
In summary, the research work presented in this work gains relevance *(i)* by expanding the literature on 3 open dimensional loading problems and *(ii)* by integrating two objective functions of interest within a lexicographic optimization.

3 The lexicographic container loading problem

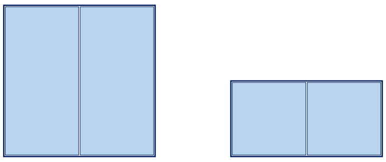
The goal of the model is to minimize the total number of used containers and pack each container effectively. As their dimensions are decision variables, here effectiveness means to minimize the volume of each container. In other words, among all feasible box loads minimizing the number of containers, our goal is to select the one minimizing their total volume.

To motivate the objective described above, three observations are in order. First, solely minimizing the number of containers can lead to undesirable solutions where, for example, boxes of various dimensions are allocated to the same container, resulting in a highly ineffective packing in terms of volume. Note that such an objective is "blind" to the boxes distribution among containers which again can result in a large wasted volume. Second, only considering to minimize the total volume of all containers is inappropriate. Such an objective could lead to the extreme situation where each container would be of the size of each box and it would only pack one box. Third, it is not of practical interest to model a biojective problem given by the two objectives just described: the shoe manufacturer's primary aim is to use the minimal number of containers, while minimizing the associated volume. To illustrate the type of solutions we are looking for, we present the following example:

Example 1. *Consider 2 boxes of size* 1×1 *and 2 boxes of size* 1×2 *to be packed into containers with length (L) and width (W) less or equal than* 2*. One possible solution minimizing the number of containers uses two of them and packs in each container* 1 *box of size* 1×1 *and* 1 *box of size* 2×1*. This distribution of boxes amongst two containers is illustrated below.*

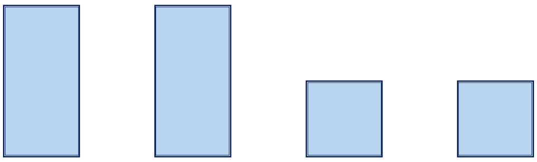


However, using two containers, it is possible to decrease the total volume of the containers, loading in one container 2 *boxes of size* 1×1 *and the other container packs two boxes of size* 2×1*. This is illustrated below.*



The two packings above minimize the number of containers. However, our goal is to be able to choose the second solution which has less total volume.

On the other hand, the following drawing illustrates the case in which we focus only on obtaining a solution with minimum volume. Such a solution uses 4 containers, one box per container.



In this context, as we will see, a lexicographic mathematical program gives a natural formulation for our multi-container loading problem with 3-open dimensions.

In lexicographic optimization, a set of objective functions is ranked, reflecting the order by which they must be optimized. In other words, if multiple optimal solutions exist for the objective ranked as first, then the selection among those solutions is done by following the order of the remaining optimization criteria. Therefore, this paradigm provides a direct way to formulate the loading problem at hand.

In our context, there are two objective functions that we wish to minimize by deciding the size of the containers, the distribution of boxes amongst the containers and the position/orientation of each box inside of each container. The objective ranked as first is the total number of containers, whereas the second is the total volume of the containers. We remark that the formulation proposed will be a mixed-integer nonlinear program for the second objective. The nonlinear part follows from the algebraic expression of the containers' volume. Next, we formalize our optimization model.

The parameters n and m designate the number of boxes to pack and the total number of available containers, respectively. We can set $m = n$ since, in the worst case, there is one box in each container. The decision variables are the following:

- β_k taking value 1 if container k is used, 0 otherwise;
- α_{ik} taking value 1 if box i is packed into container k, 0 otherwise;
- L_k, W_k, and H_k are the length, width, and height of container k, respectively;
- x_i, y_i, and z_i are the centroid's position of box i;
- p_i is the orientation of box i.

Our problem constraints include the fundamental restrictions of a container loading problem (cf. Chen et al. (1995) and Zhao et al. (2016)). The lexicographic container loading problem LCLP is given by the following mathematical program where the optimization must follow the lexicographic order of the objectives:

$$
\begin{aligned}
\text{(LCLP)} \quad \min_{\beta,\alpha,L,W,H,x,y,z,p} \quad & (f_1(\beta), f_2(H,L,W)) && \\
\text{s.t.} \quad & \beta_k \in \{0,1\}, \qquad k = 1,\ldots,m && \text{(1a)}\\
& \alpha_{ik} \le \beta_k, \quad i = 1,\ldots,n,\ k = 1,\ldots,m, && \text{(1b)}\\
& \sum_{k=1}^{m} \alpha_{ik} = 1, \quad i = 1,\ldots,n, && \text{(1c)}\\
& \text{all boxes must be placed within the assigned container,} && \text{(1d)}\\
& \text{boxes assigned to same container cannot overlap,} && \text{(1e)}\\
& \text{client constraints,} && \text{(1f)}\\
& \text{variables domain,} && \text{(1g)}
\end{aligned}
$$

where

$$f_1(\beta) = \sum_{k=1}^{m} \beta_k \tag{2}$$

$$f_2(H,L,W) = \sum_{k=1}^{m} H_k L_k W_k. \tag{3}$$

The objective function (2) corresponds to the number of containers used, while the second objective (3) is equal to the total volume of the containers. Constraints (1b) ensure that each box can only be assigned to a used container. The loading of each box to a container is enforced by constraints (1c). Constraints (1d)-(1g) will be explicitly described later. Remark that, with our lexicographic formulation, for the instance of Example 1, we would obtain the desired solution.

Equivalently, we can write LCLP as two optimization problems to be solved sequentially:

$$\begin{aligned} (\text{CLP}_1) \qquad f_1^* = \min_{\beta,\alpha,L,W,H,x,y,z,p} \quad & \sum_{k=1}^{m} \beta_k \\ \text{s.t.} \qquad & (1\text{a}) - (1\text{g}). \end{aligned}$$

$$\begin{aligned} (\text{CLP}_2) \qquad \min_{\beta,\alpha,L,W,H,x,y,z,p} \quad & \sum_{k=1}^{m} H_k L_k W_k \\ \text{s.t.} \qquad & \sum_{k=1}^{m} \beta_k = f_1^* \qquad\qquad (5\text{a}) \\ & (1\text{a}) - (1\text{g}). \end{aligned}$$

As expected and detailed below, the variables β dictate the feasibility of constraints (1d)-(1g) and thus, they cannot be discarded from CLP$_1$. The CLP$_2$ minimizes the volume of each container (3) such that no more containers than the ones decided in CLP$_1$ are used (constraint (5a)).

Next, we detail constraints (1d)-(1g). To this end, let us start by defining the parameters used by them as well as additional decision variables. Let l_i, w_i, h_i designate the length, width, and height of box i, respectively. Since our ultimate goal is to provide decision-aid to the shoe manufacturer, in what follows, certain parameters are speficied for the application at hand. Constraints (1d) and (1e) contain a big-M parameter. We set it to $M = 80$, the value of the longest edge of a container as stated in our client constraints. Note however, that such edge upper bounds are expected for other potential client constraints, simplifying the big-M value determination. The binary variables X_{ij}, Y_{ij}, and Z_{ij}, omitted in the LCLP, will be auxiliary variables establishing the relative position between boxes i and j, and preventing their overlap.

- **All boxes must be placed within the assigned container:**

$$x_i + \frac{1}{2}(p_i w_i + (1-p_i) l_i) \leq W_k + M(1-\alpha_{ik}), \tag{6}$$

$$x_i - \frac{1}{2}(\alpha_i w_i + (1-p_i) l_i) \geq 0 - M(1-\alpha_{ik}),\ i = 1,\ldots,n, \tag{7}$$

$$y_i + \frac{1}{2}(p_i l_i + (1-p_i) w_i) \leq L_k + M(1-\alpha_{ik}), \tag{8}$$

$$y_i - \frac{1}{2}(p_i l_i + (1-p_i) w_i) \geq 0 - M(1-\alpha_{ik}),\ i = 1,\ldots,n, \tag{9}$$

$$z_i + \frac{1}{2} h_i \leq H_k + M(1-\alpha_{ik}),\ z_i - \frac{1}{2} h_i \geq 0,\ i = 1,\ldots,n. \tag{10}$$

This set of constraints determine the absolute position of the boxes inside each container, given by variables x_i, y_i, z_i and orientation p_i (see Figure 1). At the same time, they ensure that boxes may only be placed with their edges parallel to the edges of the container.

- **Boxes assigned to the same container cannot overlap:**

$$x_j - x_i \geq \frac{1}{2}\left(w_i p_i + w_j p_j + l_i(1-p_i) + l_j(1-p_j)\right) - M(3 - X_{ij} - \alpha_{ik} - \alpha_{jk}),\ 1 \leq i < j \leq n, \quad (11)$$

$$x_i - x_j \geq \frac{1}{2}\left(w_i p_i + w_j p_j + l_i(1-p_i) + l_j(1-p_j)\right) - M(3 - X_{ji} - \alpha_{ik} - \alpha_{jk}),\ 1 \leq i < j \leq n, \quad (12)$$

$$y_j - y_i \geq \frac{1}{2}\left(l_i p_i + l_j p_j + w_i(1-p_i) + w_j(1-p_j)\right) - M(3 - Y_{ij} - \alpha_{ik} - \alpha_{jk}),\ 1 \leq i < j \leq n, \quad (13)$$

$$y_i - y_j \geq \frac{1}{2}\left(l_i p_i + l_j p_j + w_i(1-p_i) + w_j(1-p_j)\right) - M(3 - Y_{ji} - \alpha_{ik} - \alpha_{jk}),\ 1 \leq i < j \leq n, \quad (14)$$

$$z_j - z_i \geq \frac{1}{2}(h_i + h_j) - M(3 - Z_{ij} - \alpha_{ik} - \alpha_{jk}),\ 1 \leq i < j \leq n, \quad (15)$$

$$z_i - z_j \geq \frac{1}{2}(h_i + h_j) - M(3 - Z_{ji} - \alpha_{ik} - \alpha_{jk}),\ 1 \leq i < j \leq n, \quad (16)$$

$$X_{ij} + X_{ji} + Y_{ij} + Y_{ji} + Z_{ij} + Z_{ji} \geq \alpha_{ik} + \alpha_{jk} - 1,\ 1 \leq i < j \leq n. \quad (17)$$

These constraints guarantee that boxes assigned to same container do not overlap and for that, constraints (17) say that it is enough to enforce separation in one direction: the right-hand-side is equal to 1 if $\alpha_{ik} = \alpha_{jk} = 1$.

- **Client constraints:**

$$L_k \leq 80,\ k = 1\ldots,m, \quad (18)$$

$$W_k,\ H_k \leq 60,\ k = 1\ldots,m, \quad (19)$$

$$2(H_k + W_k) + L_k \leq 300,\ k = 1\ldots,m, \quad (20)$$

$$W_k, H_k \leq L_k,\ k = 1\ldots,m. \quad (21)$$

- **Variables domain:**

$$W_k, L_k, H_k \geq 0,\ x_i, y_i, z_i \geq 0,\ k = 1\ldots,m,\ i = 1,\ldots,n, \quad (22)$$

$$X_{ij}, Y_{ij}, Z_{ij}, \alpha_{ik}, p_i \in \{0,1\},\ i,j = 1,\ldots,n,\ i \neq j,\ k = 1,\ldots,m. \quad (23)$$

Constraints (6)-(17) are taken from Vieira et al. (2021). We give a short note on the constraints imposed by clients, (18)-(20). The manufacturer clients require shoe boxes to be facing up for label inspection and allow them to rotate. Figure 1 shows the only possible rotations of the shoe boxes. These are also preferably packed in one level for easy inspection. As a result of the client's requirements, the size of the containers should satisfy the following constraints:

1. length (L_k), considered as the longest edge of the container does not exceed 80 cm;

2. width (W_k) and height (H_k) of the container cannot exceed 60 cm;

3. $2(H_k + W_k) + L_k$ does not exceed 300 cm.

The constraint (21) is not really a client constraint. It just imposes L_k as the longest side of the container, which removes some symmetry from the problem. It must be noted that the methodologies we will propose can be adapted to handle other client constraints. However, for the sake of simplicity, we develop the presentation of this paper with the shoe manufacturer's specific constraints.

To conclude the discussion on the model, we provide the following trivial observation, allowing for eliminating the variables β from CLP$_2$ and motivate our methodology:

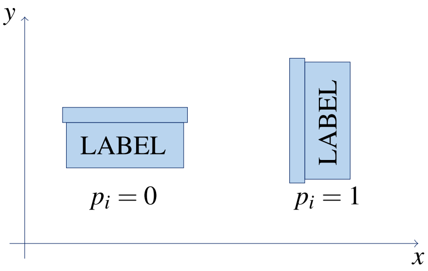


Figure 1: Allowed rotation.

Observation 1. *Let $m^\star$ be the optimal value of CLP$_1$. Then, solving the CLP$_2$ with $\beta_1, \ldots, \beta_{m^\star}$ equal 1 and the remaining equal to zero provides an optimal solution to the LCLP.*

The result above gives intuition on the theoretical intractability of LCLP, formalized below.

Proposition 1. *Consider LCLP with the client constraints replaced by generic bounds on the length, width, and height of the containers. This problem is NP-hard.*

Proof. The 3D Bin Packing is a NP-hard since it generalizes the classical Bin Packing problem which is NP-hard (Garey and Johnson , 1990). In the 3D Bin Packing problem, one seeks to pack 3D boxes in the minimum number of containers. It is easy to see that 3D Bin Packing is a special case of LCLP implying that the latter is NP-hard. Indeed, we can transform an instance of the 3D Bin Packing into an instance of the LCLP. Start by setting the lower and upper bounds of the variables H_k, L_k, and W_k equal to container k's size in the 3D Bin Packing. In this way, the second objective can be dropped since its values is constant. Then, LCLP becomes a 3D Bin Packing. □

4 A parametric mathematical program

In this section, the lexicographic container loading problem is reduced to non-linear mixed-integer parametric programs. Here, the parametrization is done by considering the β variables as parameters of CLP$_2$.

Thus, given K, the number of containers, we have the following parametric program:

$$\begin{aligned}
\min \quad & \sum_{k=1}^{m} H_k L_k W_k \\
\text{s.t.} \quad & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(CLP}_2\text{-r)} \\
& \beta_i = 1,\; i = 1, \ldots, K \\
& \beta_i = 0,\; i = K+1, \ldots, m \\
& \alpha_{ik} \le \beta_k,\; i = 1, \ldots, n,\; k = 1, \ldots, m \\
& \sum_{k=1}^{m} \alpha_{ik} = 1,\; i = 1, \ldots, n. \\
& \text{all boxes must be placed within the assigned container;} \\
& \text{boxes assigned to the same container cannot overlap;} \\
& \text{client constraints;} \\
& \text{variables domain.}
\end{aligned}$$

Note the difference between CLP$_2$-r with CLP$_2$: in CLP$_2$-r, the β variables are fixed such that $\sum_{k=1}^{m} \beta_k = K$ and it is not necessary that $K = f_1^*$. Indeed, if K is the smallest value such that CLP$_2$-r is still feasible, then by Observation 1 an optimal solution of CLP$_2$-r is also an optimal solution for the LCLP. In this way, both problems, LCLP and CLP$_2$-r, would provide a solution with minimum volume, between all the possibilities of using the least number of containers.

In other words, let K be the smallest value such that CLP$_2$-r is still feasible. Then an optimal solution of CLP$_2$-r is also an optimal solution of LCLP. We conclude that finding the optimal solution to LCLP can be done by solving at most m mixed-integer nonlinear programming problems, i.e., solving CLP$_2$-r for $K = 1,\ldots,m$. This will be algorithmically formalized in Section 6.

5 Bounds for the lexicographic container loading problem

In this section, we present lower bounds for the CLP$_1$ and the CLP$_2$. We also present an upper bound for the CLP$_2$.

We observe that the largest container is obtained by maximizing its volume subject to the client's constraints:

$$\begin{aligned}
\max \quad & H \cdot L \cdot W \\
\text{s.t.} \quad & \\
& L \leq 80 \\
& W,\ H \leq 60 \\
& 2(H+W) + L \leq 300, \\
& W, H \leq L \\
& W, L, H \geq 0.
\end{aligned}$$

An optimal solution is $H^* = 55, L^* = 80, W^* = 55$ and the optimal volume is $V^* = 242,000$ cm^3. Remark that the problem above can be directly adapted for different client constraints.

Using the volume of the largest container, we consider the classical bin-packing problem,

$$\begin{aligned}
\min \quad & \sum_{k=1}^{m} \beta_k && \text{(BPP)} \\
\text{s.t.} \quad & \sum_{i=1}^{n} v_i \alpha_{ik} \leq 242000 \beta_k,\ k = 1,\ldots,m, && (24) \\
& \sum_{k=1}^{m} \alpha_{ik} = 1,\ i = 1,\ldots,n, \\
& \alpha_{ik}, \beta_k \in \{0,1\},\ i,j = 1,\ldots,n,\ i \neq j,\ k = 1,\ldots,m,
\end{aligned}$$

where the volume of a box is $v_i = h_i \cdot l_i \cdot w_i$. This classical bin-packing problem decides the distribution of items between a minimum number of bins. The following result gives a lower bound for the CLP$_1$.

Observation 2. *The optimal value of BPP is a lower bound for the least number of containers given by the optimal solution of the CLP*$_1$.

The following proposition gives a trivial lower bound for the CLP$_2$ which we will use in the evaluation of the computational experiments.

Proposition 2. *For any choice K of the number of containers such that there is a feasible arrangement of the boxes, the following inequality holds*

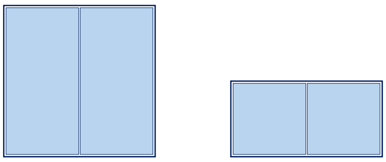
$$\sum_{i=1}^{n} v_i \leq \sum_{k=1}^{m} H_k^* L_k^* W_k^*,$$

where H_k^*, L_k^*, and W_k^* solve the CLP$_2$ with the right-hand-side of constraint (5a) *equal to K.*

Furthermore, this lower bound is tight even if $K < n$.

Proof. The inequality follows from the fact that $\sum_{i=1}^{n} v_i$ is the total volume of the boxes to be packed.

It is easy to see that the lower bound is attained if, for instance, K is set to n, i.e., K is equal to the number of boxes. However, the bound can also be tight when the number of containers is less than n, as the following example illustrates. Let us consider 2 boxes of size 1×1 and 2 boxes of size 1×2 to be pack in a container satisfying $L, W \leq 2$. An optimal solution for the LCLP is given by



where the total volume of the containers equals the total volume of the boxes. □

A solution of BPP gives a distribution α of the boxes amongst the containers. If this α is fixed in the CLP$_2$, we can disaggregate it in K independent problems where K is the number of containers to which at least one box was assigned:

$$\begin{aligned} \min \quad & H_k L_k W_k \\ \text{s.t.} \quad & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(k-fCLP}_2\text{)} \\ & \text{all boxes must be placed within the container } k; \\ & \text{boxes assigned to the container } k \text{ cannot overlap;} \\ & \text{client constraints for container } k; \\ & \text{variables domain.} \end{aligned}$$

Each independent problem has a non-linearity, which we remove by replacing the volume objective function with the linear objective $H_k + L_k + W_k$. Although this is not the volume of the container, Vieira et al. (2021) showed that this is an appropriate strategy when compared with a piecewise linear approximation of the volume, from Tsai et al. (2015) and Junqueira and Morabito (2017). Indeed, the piecewise linear approximation of the volume adds new binary variables to the problem, while simply using the linear objective, $H_k + L_k + W_k$, allows to obtain good solutions faster. Nevertheless, if given enough time, the linear approximation can obtain the optimal solution of k-fCLP$_2$. It is worth noting that the linear objective is a translation of one of the lower envelopes of the trilinear volume function presented by Meyer and Floudas (2004).

6 Algorithms for the lexicographic container loading problem

We start this section by describing an exact algorithm for the LCLP. The algorithm we present finds the optimal solution by solving a sequence of mathematical programs. First, the bin-packing problem BPP is solved. As we saw before, this gives a lower-bound on the number of containers needed to pack the boxes. Next, we solve CLP$_2$-r using the number of containers decided with BPP as K. If CLP$_2$-r is feasible, we found an optimal solution for the LCLP, if not we increase K by one and repeat the process. This is Algorithm 1.

```
Algorithm 1: Lexicographic container loading algorithm
  input  : a set of boxes to pack amongst several containers
  output: minimum number of containers, the minimum volume of each container, and the
          position of each box within the assigned container

  Solve (BPP) and save as $K$ the obtained optimal objective value;
  repeat
      solve the parametric problem CLP$_2$-r, with $K$;
      let K:=K+1;
  until (CLP$_2$-r) has solution;
  Return solution;
```

By sequentially solving the parameterized problem (CLP$_2$-r) we obtain an optimal solution, as we see below. We further note that this nonlinear mixed-integer problem can for instance be solved through the linearization of the volume given by Junqueira and Morabito (2017). However, in practice, such linearization is heavy given that multiple containers are simultaneously considered.

As the number of the boxes to be packed is finite and we do not need more containers than boxes ($K \leq n$), the algorithm terminates in a finite number of steps.

This algorithm is too expensive to be implemented, therefore we present an alternative, which finds high quality solutions. For our real-world instances, the mathematical program CLP$_2$-r is too heavy to be solved in any given computational time. Note that besides requiring to solve multiple nonlinear programs, those problems also encompass the assignment of boxes to containers which may suffer from high symmetry issues. Instead, Algorithm 2 will solve BPP and sequentially solve k-fCLP$_2$ where the nonlinear and non-convex function $H_kL_kW_k$ is replaced by $H_k+L_k+W_k$. The cost of this simplification is on the fact that the obtained solution for the objective (3) is an upper bound, while the obtained value for the objective (2) is optimal. Note that k-fCLP$_2$ with the modification on the objective function is still a mixed-integer linear program and, for certain instances' size, it is hard to obtain a guaranteed optimal solution. Nevertheless, for our instances' size, 600 seconds of running time showed to be enough to obtain good solutions. Thus, we limited the running time to solve each k-fCLP$_2$ up to 600 seconds.

Since, we are not solving the parameterized program, we must figure out a way to increase the number of containers and, at the same time, to distribute the boxes between the containers. Thus, whenever constraints (24) allow more boxes for some container than we can pack, *i.e.*, we could not find a solution for some of the k-fCLP$_2$, we replace it by constraints

$$\sum_{i=1}^{n} v_i\alpha_{ik} \leq \frac{K^*}{K}242000\beta_k,\ k=1,\ldots,m, \tag{25}$$

where K^* is optimum value of BPP and K is the current number of containers. In this way, in every loop of the cycle, by diminishing the volume of each container by the proportion K^*/K, we force the increase of the number of containers.

For the size of instances used, CPLEX returns an optimal solution for BPP in at most 90 seconds. While it is not possible to obtain a guaranteed optimal solution for each k-fCLP$_2$, CPLEX can obtain good solutions for these mixed-integer linear programs. This is showed in Section 7.

Although Algorithm 2 gives good solutions for the LCLP, it fixes the boxes assignment when minimizing the number of containers. This reduces the flexibility for minimizing the volumes, resulting in a potential decrease of the quality of the solutions for the second objective (3). In fact, Algorithm 2 tends

Algorithm 2: Short lexicographic container loading algorithm

```
input  : a set of boxes to pack amongst several containers
output: the number of containers, the volume of each container, and the position of each box
        within the assigned container

solve the bin-packing problem BPP, save the obtained $\alpha$, set $K = K^*$ containers;
for $k = 1$ to $K$ do
    solve the problem k-fCLP$_2$ with $\alpha$ fixed above, within 600 seconds of running time;
while at least one of the k-fCLP$_2$ has no solution do
    K:=K+1;
    solve the bin-packing problem BPP replacing constraints (24) by (25) and save the obtained
      $\alpha$;
    for $k = 1$ to $K$ do
        solve the problem k-fCLP$_2$ with $\alpha$, within 600 seconds of running time;
Return solution;
```

to find solutions where different boxes are allocated to the same container, while a container might be better loaded if the boxes are all equal, as illustrated in Example 1. Thus, to make an effort to improve the objective for the CLP$_2$, we consider the following constraints to be added to BPP, where r is the number of containers packing different boxes.

$$\sum_{k=1}^{m-r} h_i\alpha_{ik} + h_j\alpha_{jk} \leq 2\min\{h_i, h_j\}, \quad i, j = 1, \ldots, n, \ i < j, \tag{26}$$

$$\sum_{k=1}^{m-r} l_i\alpha_{ik} + l_j\alpha_{jk} \leq 2\min\{l_i, l_j\}, \quad i, j = 1, \ldots, n, \ i < j. \tag{27}$$

Most of the boxes can be distinguished by their heights and thus, constraints (26) will force $m-r$ containers to be loaded with boxes of the same height. In case there are different boxes of the same height, constraints (27) will then force equal boxes into $m-r$ containers, as boxes' height and length are sufficient to distinguish different boxes. Thus, these constraints allow to have at most r containers with different boxes. This is more restrictive for the CLP$_1$ objective. Nevertheless, in practice, it turns out that it did not decrease the quality of the solutions of the CLP$_1$ and it decreased the total volume of the containers, *i.e.*, the CLP$_2$ objective was improved. For small values of r, BPP can be infeasible which requires the modification of the initialization step of Algorithm 2. The value of the parameter r is chosen to be the number of different type of box and we increase it, until BPP is feasible. Thus, we obtain Algorithm 3.

7 Computational results

In this section, we report the computational experiments of Algorithms 2 and 3 with three sets of instances. The first set of instances are real orders provided by the manufacturer, named mo1, mo2, mo3, and mo4. For these ones we know the solution implemented by the manufacturer in terms of number of containers. The second set is composed with randomly generated instances for the purpose of this work (instances ro1, ro2, ro3, and ro4) and the third set contains instances co1, co2, co3 and co4. These are "controlled" instances in the sense that we know the optimal solution. For each controlled instance only one type of box is chosen and the number boxes is such that we can obtain the optimal solution packing by rows. This is related with the pallet loading problem, see Silva et al. (2016).

Algorithm 3: Lexicographic homogeneous-container loading algorithm

```
input  : a set of boxes to pack amongst several containers
output: the number of containers, the volume of each container, and the position of each box
        within the assigned container

$r$ :=number of different types of boxes;
repeat
    solve the bin-packing problem BPP in addition with constraints (26)-(27), set $K = K^*$
      containers;
    $r := r+1$;
until (BPP) is feasible;
for $k = 1$ to $K$ do
    solve the $k$ problem k-fCLP$_2$, within 600 seconds of running time;
while at least one of the k-fCLP$_2$ has no solution do
    K:=K+1;
    solve BPP replacing constraints 24 by (25) and adding constraints (26)-(27);
    for $k = 1$ to $K$ do
        solve the $k$ problem k-fCLP$_2$, within 600 seconds of running time;
Return solution;
```

Despite the LCLP refers to a 3-dimensional packing, the implementation of the algorithm is done in 2-open dimensions. In Vieira et al. (2021) a mixed-integer linear program, which is equal to k-fCLP$_2$, was tested with up to 20 boxes and it was showed that for most of the instances the containers were loaded with one level, *i.e.*, boxes were not placed on the top of each other. Thus, for computational purposes, we set the height of each container as height of tallest box in it. As consequence, instead of using the volume to evaluate the solutions, we use the bottom area. Even if we have some free space at the top of the container (caused by boxes with different heights), if the used area is close to the bottom area of the container, then the cargo can remain stable. As also argued in Vieira et al. (2021), the used bottom area is a better performance measure than the used volume.

The algorithms are implemented using the Java language and the mathematical programs are solved with CPLEX (version 12.8.0.0) accessed via the modelling language AMPL. The computations were performed on a dual core Intel(R) Xeon(R) X5675 @ 3.07 GHz with 12 Gb of memory.

We finish this introduction to the experiments by anticipating the elements impacting the efficiency of our approaches. In fact, the overall computation time does not vary much with the size of the instance. This is because solving BPP is very fast (in general less than 90 seconds), and, although solving $k-$ fCLP$_2$ can be increasingly slower with the increasing number of small boxes for the same container, our implementation limits it to 600 seconds. Remark that in Algorithms 2 and 3, the number of iterations of the while loop is given by the optimal value of CLP$_1$ minus its lower bound plus one. By consulting Table 5, we see that the while loop in the Algorithms 2 and 3 took, in the worst case, 3 iterations. Therefore, the aspects that most impact the running time are: (1) how long it takes to find a $k-fCLP_2$ with no solution within 600 seconds, remarking that this time limit is independent of the size of the instance; (2) the total number of containers which, in this case, is dependent of the instant size. The latter aspect is however bounded by 600 seconds times the number of containers.

7.1 Instances description

The possible choices of shoe boxes to pack are taken from the list in Table 1.

Table 1: Available shoe boxes.

Boxes sizes ($h \times l \times w$)		
18×12×8	27.5×17.5×10.5	30×30×11
19×13×8.5	19×20×11	32×25×11
23×12×9	23×23×11	36×25×11
21×14×9	27×23×11	34×30×11
23.5×16.5×9.2	29×23×11	40×30×11
28×13×9	29×27×11	43×30×10
25×15×9.5	29.5×13×10	45×32×11
25×17×9.5	30×18×10.5	

The first set of instances is named mo$i$, with $i = 1,2,3,4$. They were provided by the manufacturer and the first objective solutions can be compared against the manufacturer decision. This instances are described in Table 2. Instance mo1 is an order of 256 pairs of shoes, which corresponds to 14 boxes of the type 21 × 14 × 9, 140 of 23.5 × 16.5 × 9.2, and 102 of 27.5 × 17.5 × 10.5. The remaining instances read in the same way.

Table 2: Instances provided by the manufacturer

Instance	Size	Content
mo1	256	14 × (21 × 14 × 9) 140 × (23.5 × 16.5 × 9.2) 102 × (27.5 × 17.5 × 10.5)
mo2	268	17 × (21 × 14 × 9) 146 × (23.5 × 16.5 × 9.2) 105 × (27.5 × 17.5 × 10.5)
mo3	100	11 × (36 × 25 × 11) 52 × (40 × 30 × 11) 37 × (43 × 30 × 11)
mo4	252	6 × (32 × 25 × 11) 101 × (36 × 25 × 11) 105 × (40 × 30 × 11) 40 × (43 × 30 × 10)

The second set, the random instances, named ro$i$, are randomly generated in the following way: the size of the order is uniformly generated between 150 and 300, and the number of different boxes is a uniformly generated number between 2 and 4. These are listed in Table 3.

The previous called controlled instances were created in such way that we know the optimal values for both the CLP$_1$ and the CLP$_2$, and consequently for the lexicographic container loading problem (1). Just one type of box is chosen for each order and the type of box is chosen such that packing by rows, we obtain the maximum number of boxes that we can pack into the largest container. For example, order co1 contains 252 boxes of size 29.5 × 13 × 10. The largest container can at most pack 36 boxes of this type in one level. Therefore, the minimum number of containers to pack this order is $252/36 = 7$. Instance co2 differs from instance co3 because the former needs $300/50 = 6$, while the latter needs $265/50 = 5.3$ containers. This means that the sixth container could possibly pack more boxes. These instances are

Table 3: Random instances

Instance	Size	Content
ro1	186	63 × (36 × 25 × 11) 78 × (28 × 13 × 9) 45 × (27 × 23 × 11)
ro2	269	72 × (23.5 × 16.5 × 9.2) 197 × (29 × 27 × 11)
ro3	163	110 × (29.5 × 13 × 10) 32 × (19 × 20 × 11) 21 × (45 × 32 × 11)
ro4	245	79 × (19 × 13 × 8.5) 13 × (32 × 25 × 11) 126 × (27.5 × 17.5 × 10.5) 27 × (40 × 30 × 11)

listed in Table 4.

Table 4: Controlled instances

Instance	Content
co1	252 × (29.5 × 13 × 10)
co2	300 × (18 × 12 × 8)
co3	265 × (18 × 12 × 8)
co4	210 × (40 × 30 × 11)

7.2 Results for Algorithm 2

Table 5 presents the results obtained using Algorithm 2. The number of containers is given by the CLP_1 objective function and is close to the lower bound. It is one or two units above the lower bound and we remind that the lower bound is obtained using a classical bin-packing problem, which only affirms that the bins are enough to fit all the boxes, but this does not mean that one can really pack all the boxes with that number of bins. The total volume of the containers is in general less than 10% above the lower bound. The algorithm obtains the worst results for the controlled instances (co*i*) and, in this case, we are comparing the algorithm solutions with the optimal values of the CLP_1 and of the CLP_2.

The worst results were obtained with the controlled instance co2. The reason is not the size of the order, but the number of boxes loaded into each container. Most of the containers were loaded with 39 or 40 boxes. As the CLP_2 used in the algorithm is an MILP, at a certain size it is no longer possible to obtain good solutions within 600 seconds of running time. In the opposite direction is instance co4, for which we obtained the optimal solution. In this case, each container loads 14 boxes and, for this size instance, it is possible to obtain an optimal solution.

For the instances mo1, mo2, mo3, and mo4, we have some comparable results provided by the manufacturer. We compare them in Table 6 against our solutions for the lexicographic container loading problem. It can be observed that the improvement on the number of containers is significant.

Table 5: Results for Algorithm 2

Instance	CLP_1 Lower bound	Solution	CLP_2 Lower bound	Total used area	Gap	Time (h)
mo1	9	11	41758.5	44294.4	6.1%	3.0
mo2	10	11	43598.6	46401.0	6.4%	2.5
mo3	7	8	32395	33790	4.3%	1.5
mo4	16	18	76075	79090	4.0%	5.0
co1	7	8	32760	35040	7.0%	2.0
co2	6	8	28800	32400	12.5%	2.2
co3	6	7	25440	27328	7.4%	1.5
co4	15	15	69300	69300	0.0%	1.5
ro1	8	10	37836	39747	5.1%	3.5
ro2	15	17	69438.6	72945.2	5.0%	3.0
ro3	6	7	28732	30550	6.3%	1.4
ro4	10	12	44367	48012.5	8.2%	2.5

Table 6: Algorithm 2 versus manufacturer packing

Instance	Number of containers lexicographic solution	manufacturer solution
mo1	11	13
mo2	11	14
mo3	8	10
mo4	18	26

7.3 Comparison between Algorithms 2 and 3

The relevance of comparing both algorithms and not just testing Algorithm 3, is to observe if the number of containers can be increased by this algorithm. Thus, Table 7 compares the results of both algorithms, 2 and 3. We then observe that while the introduction of constraints (26) and (27) does not worsen, in general, the first objective (2), it improves the second objective (3), by decreasing the total volume of the containers. The best results are emphasized in bold. The gap with Algorithm 3 is generally improved and it is around 5% or below.

Table 7: Comparison between Algorithms 2 and 3

Instance	CLP_1 Algorithm 2	Algorithm 3	CLP_2 Algorithm 2	Algorithm 3	New gap
mo1	11	11	44294.4	**43556.35**	4.3%
mo2	**11**	12	46401.0	**45810.9**	5.1%
mo3	8	8	**33790**	33900	4.6%
mo4	18	18	79090	**78800**	3.6%
ro1	10	**9**	39747	**39564**	4.6%
ro2	17	17	72945.2	**70345.9**	1.3%
ro3	7	7	30550	**30279**	5.4%
ro4	12	12	48012.5	**46847**	5.6%

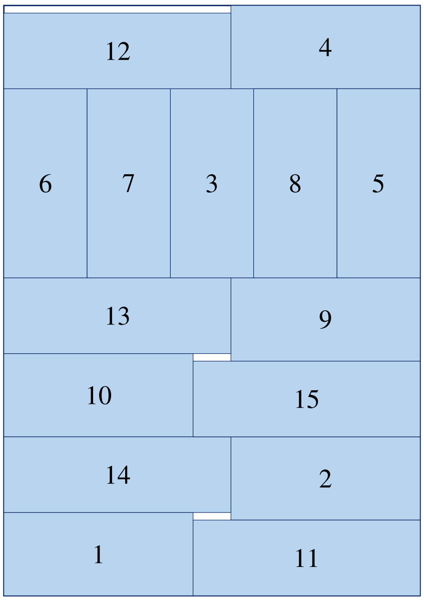


Figure 2: Layout of one container from instance mo4

As the controlled instances have only one type of boxes, the mentioned constraints do not change anything and thus they are not tested.

We finish this section with an illustration in Figure 2 of one container of a solution to instance mo4, with fifteen shoe boxes.

8 Conclusion

In this paper, it was dealt for the first time a container loading problem with a hierarchy for the objectives to be optimized. Although the theoretically intractability of the problem, we explored its specific features to design a specialized algorithm where a solution from the CLP$_1$ feeds the CLP$_2$. We also introduced a restrictive variant of the algorithm, which improves the quality of the second optimization criterion, without losing the quality of the first objective. This is relevant because we increased the stability of the cargo. More importantly, from a practical point of view, if containers are not tight to the cargo they contain shoe boxes might open and shoes can spread all over the container. The latter scenario is not appreciated by the manufacturer clients. Computational results showed that the algorithms are efficient to obtain good results when compared with the given lower bounds.

As future work, there are natural extensions that can directly benefit from the formulation presented in this paper. For example, it would be of interest to seek ways of homogenizing the containers, by introducing a classification of boxes according with their size. Further improvements of our solution technique are of interest too. A critical issue is the symmetry of solutions when assigning boxes to containers and when loading boxes in each container. While our heuristics avoid symmetry issues at the boxes' assignment level, we did not tackle the loading symmetries for each container.

Acknowledgments

The first author was partially supported by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the project UI/297/2020 (Centro de Matemática e Aplicações).

The second author thanks the support of the Institut de valorisation des données and Fonds de recherche du Québec through the FRQ–IVADO Research Chair in Data Science for Combinatorial Game Theory, and the Natural Sciences and Engineering Research Council of Canada through the discovery grant 2019-04557.

References

Alonso MT, Alvarez-Valdes R, Iori M, Parreño F, Tamarit JM (2017) Mathematical models for multicontainer loading problems. Omega 66(Part A):106-117

Araya I, Moyano M, Sanchez C (2020) A beam search algorithm for the biobjective container loading problem. European Journal of Operational Research 286(2), 417-431

Bortfeldt A, Wäscher G (2013) Constraints in container loading - A state-of-the-art review. European Journal of Operational Research 229(1):1-20

Coniglio S, D'Andreagiovanni F, Furini F (2019) A lexicographic pricer or the fractional bin packing problem. Operations Reserach letters 47:622-628.

Chen CS, Lee SM, Shen QS (1995) An analytical model for the container loading problem. European Journal of Operational Research 80(1): 68-76

Dahmani N, Clautiaux F, Krichen S, Talbi E, (2013) Iterative approaches for solving a multi-objective 2-dimensional vector packing problem. Computers & Industrial Engineering 66 (1), 158-170.

Delorme M, Iori M, Martello S, (2016) Bin packing and cutting stock problems: Mathematical models and exact algorithms. European Journal of Operational Research, 255(1), pp.1-20.

Ehrgott M (2005) Multicriteria optimization. Springer-Verlag Berlin Heidelberg

Fernández A, Gil C, Baños R, Montoya M G (2013) A parallel multi-objective algorithm for two-dimensional bin packing with rotations and load balancing. Expert Systems with Applications 40(13), 5169-5180.

M E Garey, D S Johnson (1990) Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., USA.

Hessler K, Irnich S, Kreiter T, Pferschy U (2020) Lexicographic bin-packing optimization for loading trucks in a direct-shipping system. Discussion paper, Gutenberg School of Management and Economics.

Junqueira L, Morabito R (2017) On solving three-dimensional open-dimension rectangular packing problems. Engineering Optimization 49(5), 733-745

Khanafer A, Clautiaux F, Hanafi S, Talbi E (2012) The min-conflict packing problem. Computers & Operations Research 39(9), 2122-2132.

Liu Y, Chu C, Yu Y (2012) Aggregated state dynamic programming for a multiobjective two-dimensional bin packing problem. International Journal of Production Research, 50(15), 4316-4325

Liu Y, Chu C, Wu F (2014) A heuristic for variable size multiobjective two-dimensional bin packing. International Journal of Production Research 52(12), 3450-3465

Liu D S, Tan K C, Huang S Y, Goh C K, Ho W K (2008) On solving multiobjective bin packing problems using evolutionary particle swarm optimization. European Journal of Operational Research 190(2), 357-382.

Meyer C and Floudas C (2004) Trilinear Monomials with Positive or Negative Domains: Facets of the Convex and Concave Envelopes. Frontiers in Global Optimization 327-352.

Silva E F, Toffolo T A M, Wauters T (2019) Exact methods for three-dimensional cutting and packing: A comparative study concerning single container problems. Computers and Operations Research 109, 12-27.

Silva E, Oliveira J F, Wäscher G (2016) The pallet loading problem: a review of solution methods and computational experiments. International Transactions in Operational Research 23, 147-172.

Spencer K Y, Tsvetkov P V, Jarrell J J (2019) A greedy memetic algorithm for a multiobjective dynamic bin packing problem for storing cooling objects. J Heuristics 25, 1–45.

Toffolo TAM, Esprit E, Wauters T, Vanden Berghe G (2017). A two-dimensional heuristic decomposition approach to a three-dimensional multiple container loading problem. European Journal of Operational Research 257(2), 526-538

Tsai J-F, Li H-L (2006). A global optimization method for packing problems. Engineering Optimization 38, 687-700

Tsai J-F, Wang P-C, Lin M-H (2015). A global optimization approach for solving three-dimensional open dimension rectangular packing problems. Optimization 64(12), 2601-2618

Vieira MVC, Ferreira F, Duque J, Almeida R (2021) On the packing process in a shoe manufacturer. Journal of the Operational Research Society 72(4), 853-864.

Vieira MVC, Ferreira F (2020) Packing Shoes Efficiently. Impact 2020, 34-38

Zhao X, Bennell JA, Bektas T, Dowsland KA (2016). A comparative review of 3D container loading algorithms. International Transactions in Operational Research 23(1-2), 287-320

Zheng J-N, Chien C-F, Gen M (2015) Multi-objective multi-population biased random-key genetic algorithm for the 3-D container loading problem. Computers & Industrial Engineering 89, 80-87