CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

The Exact Solutions of Several Classes of Container Loading Problems

Deidson Vitorio Kurpel
Cleder Marcos Schenekemberg
Cassius Tadeu Scarpin
José Eduardo Pécora Junior
Leandro C. Coelho

June 2018

CIRRELT-2018-26

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2018-014.

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

The Exact Solutions of Several Classes of Container Loading Problems

Deidson Vitorio Kurpel[1], Cleder Marcos Schenekemberg[1], Cassius Tadeu Scarpin[1], José Eduardo Pécora Junior[1,2], Leandro C. Coelho[2,3]

1. Universidade Federal do Paraná, Rua XV de Novembre, 1299, CEP 80.060-000, Centro Curitiba, PR Brasil
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
3. Department of Operations and Decision Systems, 2325 de la Terrasse, Université Laval, Québec, Canada G1V 0A6

Abstract. In this paper we address multiple container loading problems, consisting of placing rectangular boxes, orthogonally and without overlapping, inside containers in order to optimize a given objective function, generally maximizing the value of the packed items or minimizing the number of containers required to pack all available boxes. Four techniques to enumerate the possible locations of boxes inside a container, not yet tested in literature, are developed and evaluated. We also propose new techniques to obtain bounds for these problems improving an existing heuristic method. In addition, we study some practical considerations of box orientation, load stability, and separation of boxes. A detailed analysis shows that our approach is very competitive, generating models containing signicantly fewer variables and constraints than the traditional approach existing in the literature. We test our methods on several existing benchmark sets. We prove optimality and improve the best known results for several instances.

Keywords. Container loading problem, packing, practical considerations, mathematical formulation, heuristics.

Acknowledgements. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764. This support is gratefully acknowledged. We thank Calcul Québec for providing high performance computing facilities.

---

* Corresponding author: Leandro.Coelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2018

1 Introduction

In Container Loading Problems (CLPs) the goal is to place rectangular boxes orthogonally inside a container to optimize an objective function, typically maximizing the value of the selected boxes (when the number of containers is not sufficient to load all items) or minimizing the number of containers required to pack all boxes. These problems arise in freight distribution when packing boxes inside actual containers for maritime or air transportation or in trucks for road distribution. Due to the nature of the problem, one must consider all three dimensions of the boxes and of the containers; special and simplified cases arise in one and two dimensions as well.

An arrangement of boxes inside a container is called loading pattern. A better use of the container's space can substantially reduce the cost of freight, yielding significant financial implications for carriers and shippers. Moreover, practical considerations arise when designing a loading pattern, such as the orientations of the boxes, the stability of the load, and the distribution of weight inside the containers. Bischoff and Ratcliff [1] list 12 practical considerations commonly observed in CLPs. However, most of them have not been properly studied in the literature [3].

In general, CLPs are classified into two groups of problems [25]. The first one is the *input minimization problem*, in which the storage space is sufficient to pack all boxes; here, the number of containers is usually not binding. The objective function is to minimize the number of containers required to load all available boxes. In the second group, called *output maximization problem*, the space of a limited number of containers is not sufficient to store all the boxes. The goal is then to select a subset of boxes maximizing the volume or value associated with the load.

Several approaches to solve these problems have been proposed in the literature. Ivancic et al. [14] solve the input minimization problem with a hybrid of heuristic and integer programming that fills the containers sequentially. Eley [9] presents a greedy heuristic that creates blocks of identical boxes to be packed into the container, and a tree-search heuristic to improve the loading patterns. The method of Lim and Zhang [16] uses a greedy heuristic that fills the containers sequentially, prioritizing large boxes. Both Che et al. [5] and Zhu et al. [26] use a procedure to generate columns representing packings to solve an extended set covering problem.

Output maximization problems have also attracted the interest of many researchers. Mohanty et al. [18] present a sequential solution strategy using a column generation procedure. Bortfeldt [2] uses a sequential strategy to fill the containers, outlining some strategies to select the boxes to be placed in the containers. Modifications were made to deal with input minimization problems as well. Eley [10] uses a column generation heuristic to generate a sufficient number of loading patterns and then solves an integer programming problem, considering only this limited number of patterns. Takahara [23] introduces a multi-start local search procedure to determine the best loading sequence of types of boxes and their orientation to be placed in the container. Ren et al. [21] solve several single container problems, filling each one with the cuboid arrangement approach and then improve the solution using a tree search algorithm. Junqueira et al. [15] present a 0-1 integer linear programming approach to an output maximization problem considering a single container.

The input minimization problems addressed in this paper can be classified, according to the typology of Wäscher et al. [25], as Single Stock-Size Cutting Stock Problems (SSSCSP), while the output maximization problems studied can be classified as Multiple Heterogeneous Large Object Placement Problem (MHLOPP). Recent works dealing with the SSSCSP can be found in Grunewald et al. [12]; to the best of our knowledge, no papers exploring the MHLOPP have been published since that of Ren et al. [21].

In this paper we present exact approaches to solve problems dealing with multiple containers, both for input minimization and output maximization. We also enhance several loading patterns to account for the three-dimensional issues inherent to CLPs. To generate an upper bound for input minimization problems, we solve a set cover problem. To create the columns for this problem, we use the heuristic of George and Robinson [11] and also present a variation of it, extending the work of Moura and Oliveira [19]. The 0-1 integer linear programming model presented in this work, based on the formulations for the single container loading problems of Junqueira et al. [15], allows us to determine the exact solution of problems in different scenarios, combining practical considerations in ways that have not been previously tested in the CLP literature. These practical considerations include the orientation of the boxes as well as their stability and the separation of boxes that cannot be loaded into the same container. To our knowledge, no exact formulation exists to deal with the separation of boxes.

The remainder of this paper is organized as follows. Section 2 presents 0-1 integer linear programming models, along with the practical considerations of box orientation, load stability, and box separation. In Section 3 we derive tight bounds for input minimization problems. Section 4 analyzes the extensive computational results obtained on classical benchmark instances and presents the comparison with many competing algorithms. Finally, Section 5 presents the main conclusions as well as perspectives for future work.

2 Mathematical formulations

We consider a set of m distinct types of boxes available, and each box of type $i \in \{1, \ldots, m\}$ has a length l_i, a width w_i, a height h_i, a volume (or value) v_i, and availability b_i. For input minimization problems, we consider that there are a total of C identical containers, each with length L, width W, height H and volume V. For output maximization problems, we consider $\mathcal{K}$ types of containers, each container of type $k \in \{1, \ldots, \mathcal{K}\}$ being associated with length L_k, width W_k, height H_k, volume V_k, and availability d_k.

Taking the Cartesian coordinate system, let (p, q, r) be the front-left-bottom vertex of a box inside a container (see Figure 1). Boxes can assume different orientations within the container. Although a box can be placed inside a container in up to six different orientations, more restricted situations can be addressed, in which it is not possible to place a box in a given orientation, generally due to fragility and/or stability of the cargo.

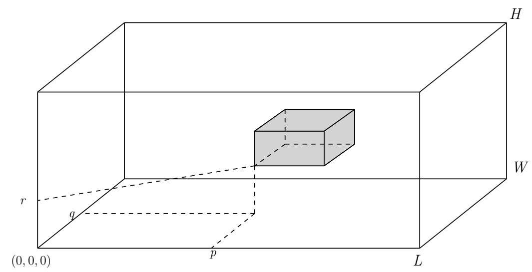

Figure 1: Box allocated inside a container with its vertex at the point (p, q, r).

In our formulations, we address the practical consideration of box orientation, assuming that a box of type i can be loaded in up to six orientations, by means of orthogonal rotations. To this end, we derive the orientations that a given box can take by decomposing the dimensions (l_i, w_i, h_i) into new items of dimensions (l_{ig}, w_{ig}, h_{ig}), where $g \in \Omega_i \neq \emptyset$, with $\Omega_i \subseteq \{1, 2, 3, 4, 5, 6\}$, such that $(l_{i1}, w_{i1}, h_{i1}) = (l_i, w_i, h_i)$, $(l_{i2}, w_{i2}, h_{i2}) = (l_i, h_i, w_i)$, $(l_{i3}, w_{i3}, h_{i3}) = (w_i, l_i, h_i)$, $(l_{i4}, w_{i4}, h_{i4}) = (w_i, h_i, l_i)$, $(l_{i5}, w_{i5}, h_{i5}) = (h_i, l_i, w_i)$ and $(l_{i6}, w_{i6}, h_{i6}) = (h_i, w_i, l_i)$, as shown in Figure 2.

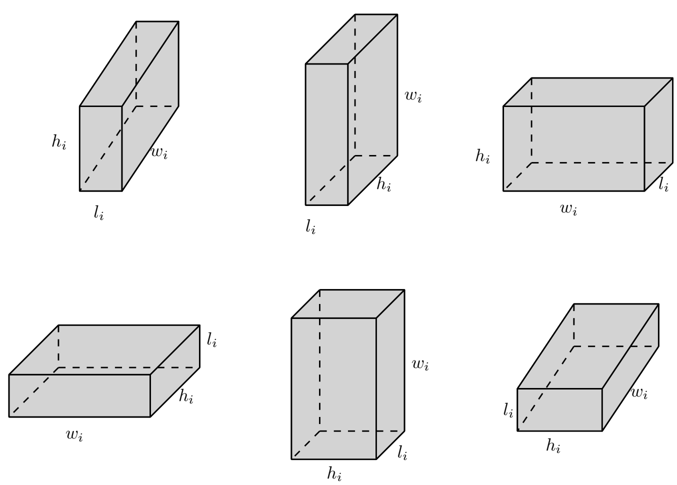

Figure 2: Six possible orientations of a box.

The following sets indicate the possible positions that a box can take in relation to the dimensions of the container:

$$X = \{p \in \mathbb{Z} \,|\, 0 \leq p \leq L - \min_i(l_{ig})\}, \quad \forall \quad i \in \{1, \ldots, m\}, \quad g \in \Omega_i \tag{1}$$

$$Y = \{q \in \mathbb{Z} \,|\, 0 \leq q \leq W - \min_i(w_{ig})\}, \quad \forall \quad i \in \{1, \ldots, m\}, \quad g \in \Omega_i \tag{2}$$

$$Z = \{r \in \mathbb{Z} \,|\, 0 \leq r \leq H - \min_i(h_{ig})\}, \quad \forall \quad i \in \{1, \ldots, m\}, \quad g \in \Omega_i. \tag{3}$$

Enumerating the sets that indicate all possible positions that a box can take along the axes of a container allow only the resolution of small problems due to the large number of possible positions that must be considered when deciding where to place an item, many of them redundant or symmetric. In this paper, four techniques to generate the possible points that can be occupied by the front-left-bottom vertex of a box are adapted for the three-dimensional packing case: Normal Patterns (NP), Reduced Raster Points (RRP), Regular Normal Patterns (RNP), and the Meet in the Middle (MiM) Principle.

The NP, originally described by Herz [13] and Christofides and Whitlock [6], take into account that the boxes can be moved toward the bottom and/or left of the container until they are adjacent to other boxes or to the container walls, eliminating symmetrical positions that they may occupy. In an attempt to obtain sets with fewer elements, Terno et al. [24] and Scheithauer and Terno [22] introduced the RRP sets, derived from NP. Although there is no guarantee that no loss of generality occurs in the RRP, in the empirical tests performed by de Queiroz et al. [8] and also in this work, no optimal solution was missed.

Boschetti et al. [4] introduce the RNP, in which the possible positions that a box i can take inside a container can be computed by determining the positions of all box types except i. Finally, the MiM Principle, defined by Côté and Iori [7], fix a threshold $\mathbb{T}$ along the dimensions of the container, forcing items whose front-left-bottom vertex lay in the left of $\mathbb{T}$ to be packed on the container's bottom, while the others are positioned at the top of the container.

These discretization techniques aim to generate sets of lower cardinality than those presented by formulation (1)–(3), leading to models with fewer variables and constraints.

2.1 Input minimization problems

In order to model input minimization problems, we use the following sets and decision variables. The sets described by (4)–(6) enumerate the possible coordinates that a box of type i, in its g^{th} orientation, can assume inside a container:

$$X_{ig} = \{p \in X \,|\, 0 \leq p \leq L - l_{ig}\}, \quad \forall \quad i \in \{1, \ldots, m\}, \quad g \in \Omega_i \tag{4}$$

$$Y_{ig} = \{q \in Y \,|\, 0 \leq q \leq W - w_{ig}\}, \quad \forall \quad i \in \{1, \ldots, m\}, \quad g \in \Omega_i \tag{5}$$

$$Z_{ig} = \{r \in Z \,|\, 0 \leq r \leq H - h_{ig}\}, \quad \forall \quad i \in \{1, \ldots, m\}, \quad g \in \Omega_i. \tag{6}$$

The decision variables of the model are defined as follows. Let x_{ig}^{jpqr} be equal to 1 if a box of type i, in its g^{th} orientation, has its front-left-bottom vertex at point (p, q, r) of the j^{th} container, and 0 otherwise, and e_j be equal to one if the j^{th} container is used, $j \in \{1, \ldots, C\}$, and 0 otherwise.

The mathematical formulation for the input minimization problem is given by:

$$\min \sum_{j=1}^{C} e_j \tag{7}$$

subject to:

$$\sum_{i=1}^{m} \sum_{g \in \Omega_i} \sum_{\{p \in X_{ig} | s - l_{ig} + 1 \leq p \leq s\}} \sum_{\{q \in Y_{ig} | t - w_{ig} + 1 \leq q \leq t\}} \sum_{\{r \in Z_{ig} | u - h_{ig} + 1 \leq r \leq u\}} x_{ig}^{jpqr} \leq e_j,$$
$$s \in X, \quad t \in Y, \quad u \in Z, \quad j \in \{1, \ldots, C\} \tag{8}$$

$$\sum_{j=1}^{C} \sum_{g \in \Omega_i} \sum_{p \in X_{ig}} \sum_{q \in Y_{ig}} \sum_{r \in Z_{ig}} x_{ig}^{jpqr} = b_i, \quad i \in \{1, \ldots, m\} \tag{9}$$

$$x_{ig}^{jpqr} \in \{0, 1\}, \quad i \in \{1, \ldots, m\}, \quad g \in \Omega_i, \quad j \in \{1, \ldots, C\}, \quad p \in X_{ig}, \quad q \in Y_{ig}, \quad r \in Z_{ig}. \tag{10}$$

The objective function (7) minimizes the number of containers used. Constraints (8) guarantee that no overlapping of boxes inside the j^{th} container occurs, if it is used. These constraints work by preventing a point (s, t, u) occupied by any item from being used by another item. Constraints (9) impose that all boxes must be packed, allowing the boxes to be rotated, assuming up to six positions, while constraints (10) define the domain of the decision variables.

Given that all containers are identical, equivalent solutions may arise by simply interchanging their usage. To avoid such symmetries, we consider the following constraints, which imposes an order of usage of the available containers:

$$e_j \leq e_{j-1}, \quad j \in \{2, \ldots, C\}. \tag{11}$$

2.2 Output maximization problems

Some adjustments in the objective function and constraints in the input minimization model allow the construction of a mathematical formulation for output maximization problems with multiple heterogeneous containers.

Since each type of container has different dimensions, in the formulation described next, sets X_k, Y_k and Z_k can be obtained repeating the process of set generation using one of the discretizations from Section 2 for each of the k types of containers available, just like sets X_{igk}, Y_{igk} and Z_{igk} can be devised from replications of the sets (4)–(6) for all $k \in \{1, \ldots, \mathcal{K}\}$.

The decision variables are defined as follows. Let x_{ig}^{jkpqr} be equal to 1 if and only if a box of type i, in its g^{th} orientation, has its front-left-bottom vertex at the point (p, q, r) of the j^{th} container of type k. The mathematical formulation for the output maximization problem is given by:

$$\max \sum_{k=1}^{\mathcal{K}} \sum_{j=1}^{d_k} \sum_{i=1}^{m} \sum_{g\in\Omega_i} \sum_{p\in X_{igk}} \sum_{q\in Y_{igk}} \sum_{r\in Z_{igk}} v_i x_{ig}^{jkpqr} \tag{12}$$

subject to

$$\sum_{i=1}^{m} \sum_{g\in\Omega_i} \sum_{\{p\in X_{igk}|s-l_{ig}+1\le p\le s\}} \sum_{\{q\in Y_{igk}|t-w_{ig}+1\le q\le t\}} \sum_{\{r\in Z_{igk}|u-h_{ig}+1\le r\le u\}} x_{ig}^{jkpqr} \le 1,$$
$$s\in X_k,\quad t\in Y_k,\quad u\in Z_k,\quad k\in\{1,\ldots,\mathcal{K}\},\quad j\in\{1,\ldots,d_k\} \tag{13}$$

$$\sum_{k=1}^{\mathcal{K}} \sum_{j=1}^{d_k} \sum_{g\in\Omega_i} \sum_{p\in X_{igk}} \sum_{q\in Y_{igk}} \sum_{r\in Z_{igk}} x_{ig}^{jkpqr} \le b_i,\quad i\in\{1,\ldots,m\} \tag{14}$$

$$x_{ig}^{jkpqr}\in\{0,1\},\quad i\in\{1,\ldots,m\},\quad g\in\Omega_i,\quad k\in\{1,\ldots,\mathcal{K}\},$$
$$j\in\{1,\ldots,d_k\},\quad p\in X_{igk},\quad q\in Y_{igk},\quad r\in Z_{igk}. \tag{15}$$

The objective function (12) maximizes the volume (or the associated value) of the items packed in the available containers. Constraints (13)–(15) have similar interpretation to those from the previous section. Note that if there is only one type of container, the formulation can be simplified by removing index k and assuming that $j \in \{1, \ldots, C\}$.

2.3 Practical considerations

In this section we study some practical considerations not often included in other approaches. Nevertheless, they are of high importance in practice. These include the separation of boxes in Section 2.3.1 and load stability in Section 2.3.2.

2.3.1 Separation of boxes

According to Bischoff and Ratcliff [1], requirements of separation of boxes are related to items that cannot be packed side by side, having some space between them when they share the same container. Eley [10] extends this concept to types that cannot be loaded into the same container. We follow this interpretation of conflicting items, i.e., they cannot be loaded into the same container.

To this end, let B_1 and B_2 be two sets with the types of conflicting items, i.e., each box type in B_1 cannot be loaded in the same container with items of set B_2 and vice-versa. A practical situation arises when one must load some containers with a set of different foods and another set of chemicals, or separating frozen and refrigerated items from general cargo. Binary variables $y_{\gamma jk}$, $\gamma \in B_1 \cup B_2$, $j \in \{1, \ldots, d_k\}$ and $k \in \{1, \ldots, \mathcal{K}\}$, take value 1 if boxes of type γ are allocated in the j^{th} container of type k, and 0 otherwise.

To consider the practical consideration of separation of items, constraints (14) must be replaced by the following ones, for each $\gamma \in B_1 \cup B_2$:

$$\sum_{g \in \Omega_\gamma} \sum_{p \in X_{\gamma gk}} \sum_{q \in Y_{\gamma gk}} \sum_{r \in Z_{\gamma gk}} x_{\gamma g}^{jkpqr} \leq b_\gamma y_{\gamma jk}, \quad k \in \{1, \ldots, \mathcal{K}\}, \quad j \in \{1, \ldots, d_k\}. \tag{16}$$

To guarantee that boxes of type $\theta \in B_1$ will be loaded in different containers than boxes of type $\mu \in B_2$, the following constraints must be added to the model defined by (12)–(15):

$$\begin{aligned} & y_{\theta jk} + y_{\mu jk} \leq 1 \\ & y_{\theta jk} \in \{0,1\}, \quad y_{\mu jk} \in \{0,1\}, \quad \theta \in B_1, \quad \mu \in B_2, \quad k \in \{1, \ldots, \mathcal{K}\}, \quad j \in \{1, \ldots, d_k\}. \end{aligned} \tag{17}$$

Again, if there is only one type of container, the formulation can be simplified by removing index k and assuming that $j \in \{1, \ldots, C\}$.

2.3.2 Stability of items

Stability is one of the most important practical considerations that can be incorporated in the CLP, as stable loads prevent cargo damage during transport and ensure the safety of operators, especially during the loading/unloading procedures. The stability requirement imposes that a minimum percentage of the base of each box must be supported by the base of the container of by other boxes. The stability coefficient α indicates the minimum portion of the items that must be supported. When $\alpha = 1$, 100% of the base of the box must be supported, and when $\alpha = 0$ no stability requirement is considered, i.e., the items may be partially supported or “floating” within the containers.

To consider the vertical stability in the CLP, the following constraints must be added to the formulations (7)–(10) and (12)–(15):

$$\sum_{i=1}^{m} \sum_{\{g \in \Omega_i | r' - h_{ig} \geq 0\}} \sum_{\{p \in X_{igk} | p' - l_{ig} + 1 \leq p \leq p' + l_{\lambda a} - 1\}} \sum_{\{q \in Y_{igk} | q' - w_{ig} + 1 \leq q \leq q' + w_{\lambda a} - 1\}} L_{i\lambda} W_{i\lambda} x_{ig}^{jkpq(r' - h_{ig})} \geq \alpha l_{\lambda a} w_{\lambda a} x_{\lambda a}^{jkp'q'r'}$$
$$\lambda \in \{1, \dots, m\}, \quad a \in \Omega_\lambda, \quad k \in \{1, \dots, \mathcal{K}\}, \quad j \in \{1, \dots, d_k\},$$
$$p' \in X_{\lambda a k}, \quad q' \in Y_{\lambda a k}, \quad r' \in Z_{\lambda a k}. \quad (18)$$

where

$$L_{i\lambda} = \min(p + l_{ig}, p' + l_{\lambda a}) - \max(p, p') \quad (19)$$

and

$$W_{i\lambda} = \min(q + w_{ig}, q' + w_{\lambda a}) - \max(q, q'). \quad (20)$$

Expressions (18)–(20) guarantee a minimum percentage of support for the base for a box of type λ positioned at (p', q', r') of the j^{th} container of type k. This support is provided by a box of type i (including $i = \lambda$) that is located at (p, q, r) within the container, where $r = r' - h_{ig}$.

As in the mathematical formulation for the practical consideration of item separation, one can remove index k and assume that $j \in \{1, \dots, C\}$ if there is only one type of container available.

We note, however, that while the parameter α controls the minimum bearing area of the boxes, it does not guarantee the stability of the load when it is subject to the action of external forces such as speed, acceleration and vehicle oscillations [20].

3 Bounds for input minimization problems

In this section we describe how we obtain bounds for input minimization problems. We describe an upper bound procedure in Section 3.1 and a lower bound method in Section 3.2. Computational results showing their efficiency are presented in Section 4.

3.1 Upper bound

Determining the maximum number of containers $\mathcal{U}$ needed to pack all the boxes is extremely important to model the input minimization problems. Since the number of containers C is a parameter in the formulation, the smaller the value of C, the smaller the number of variables and constraints of the models.

In order to obtain an upper bound for the problem, we solve a set cover problem, minimizing the number containers needed to pack all the boxes. Inspired by the work of Zhu et al. [26], we create packings to the set cover problem but, instead of using the prototype column generation presented by them, we use the heuristic of George and Robinson [11] (G&R), a fast procedure to obtain the packings. Besides, we also present a variation of the heuristic based on the GRMod of Moura and Oliveira [19].

Originally designed for loading a single container, the G&R heuristic fills the container by partitioning it into layers and filling them with stacks of identical boxes. The depth of each layer is defined by the first box placed in the layer. George and Robinson [11] present three criteria for choosing this box. These criteria, given below, must be applied sequentially, i.e., the next criterion is used only used as a tie-breaker:

i. Select the box with the largest of the smallest dimensions;

ii. Select the item with the largest quantity available;

iii. Select the box with the largest dimension.

While filling the container, it is possible that the box type chosen does not have a sufficient availability to fill the entire layer or, alternatively, that this type of box can not be placed in the remaining space. This entails the generation of new spaces inside the container: a depth space in front of the layer; a width space on the side of the layer and a height space on the top of the layer. These spaces must be filled in the reverse order in which they are created, that is, one must first try to fill the height space, then the width and, finally, the depth space. However, these spaces may be insufficient to be occupied by any other type of box during the filling of the current layer. Even so, they are stored in a list of temporarily rejected spaces, because they can be combined with other idle spaces in filling a new layer, generating a more dense loading.

To generate feasible packing patterns, we applied the heuristic sequentially, updating the availability of the boxes as each container is filled. One of the disadvantages of this strategy is that, although the first containers tend to be almost completely filled, the latter ones will have a low ratio of occupancy. This is mainly due to the number of boxes available for loading, as well as the need to choose the boxes that will open the layers based on the original heuristic criteria.

In order to obtain different, high quality packings for the set cover, we created a variation of the G&R heuristic, inspired by the work of Moura and Oliveira [19]. In order to generate the load patterns, we eliminated the criteria for choosing the boxes of the original heuristic and made the choice of the type of box that will be used to create a new layer or to fill a space based on a restricted list of candidates. The boxes that will be part of this list are chosen based on the following expression, given by Moura and Oliveira [19]:

$$\mathcal{T} = Vol_{max} + \beta(Vol_{min} - Vol_{max}) \tag{21}$$

where $\mathcal{T}$ is a volume utilization threshold, Vol_{max} and Vol_{min} are, respectively, the maximum and minimum volume that can be used by the arrangements that can be formed with the boxes available for loading, and β is a parameter that controls to dictate the level of randomness of the algorithm. Moura and Oliveira [19] point out that when $\beta = 1$, the boxes will be chosen in a random fashion, whereas when $\beta = 0$, the choice of the boxes assumes a greedy behavior. Since this approach has a random feature, each execution of it can yield a different load pattern.

One of the advantages of using a heuristic like G&R or the variation of the GRMod to obtain the packings is that, besides the number of each box type loaded in a container, the heuristic also gives an actual packing pattern, with a feasible arrangement of boxes inside a container.

After running the strategies above, we have $\mathcal{P}$ different packing patterns. The number of boxes types loaded in each packing will be a column vector y, whose elements y_j^i indicate the number of boxes of type i in the packing j. This leads to the following set cover formulation:

$$\min \sum_{j=1}^{\mathcal{P}} \delta_j \tag{22}$$

subject to:

$$\sum_{j=1}^{\mathcal{P}} y_j^i \delta_j = b_i, \qquad i = 1, \ldots, m \tag{23}$$

$$\delta_j \in \mathbb{N}. \tag{24}$$

In the formulation above, (22) minimizes the number of times a packing is selected. This can also be seen as the number of containers needed to pack the available boxes; therefore, by solving (22)–(24), an upper bound for the input minimization problem is obtained.

3.2 Lower bound

We computed a lower bound to determine the minimum number of containers needed to load the available boxes. In addition, we compare the results obtained by the upper bound procedure with the lower bound to evaluate the quality of the initial solution. An obvious lower bound for the CLP, which can determine the minimum number of containers needed to allocate the boxes, is given by:

$$\mathcal{L} = \left\lceil \frac{\sum_{i=1}^{m} v_i b_i}{V} \right\rceil . \tag{25}$$

It should be noted that it may not be possible to place all of the boxes in this number of containers, because the waste of space may be unavoidable due to the geometric arrangement of the boxes [17].

The objective function in the input minimization problem formulation minimizes the number e_j of containers needed to allocate all the available boxes. By using (25), since $\mathcal{L} \leq \mathcal{U}$, one can assign the value 1 to all variables e_j, $j \in \{1, \ldots, \mathcal{L}\}$, leaving the mathematical formulation to minimize the use of remaining variables, and to ensure the arrangement of the boxes in the selected containers, considerably reducing the effort to solve the models.

Finally, if a loading pattern is known and $\mathcal{L} = \mathcal{U}$, it constitutes an optimal solution for the problem.

4 Computational experiments

We now describe the detailed computational experiments used to assess the performance of our methods. In order to evaluate the approaches proposed in the previous sections, we have used known datasets from the literature for both input minimization and output maximization problems. We have implemented all models and procedures in VB.net and executed each instance using computers equipped with Xeon processors running at

2.77GHz and up to 120GB of RAM. A time limit of 7200 seconds was imposed for each execution. The mathematical models were solved using Gurobi 7.0.2.

Section 4.1 describes the instances used and the methods available in the literature. Section 4.2 presents a discussion about the discretizations techniques used to solve the models. In Section 4.3 we detail the results obtained for input minimization problems, followed by those for output maximization in Section 4.4.

4.1 Instances

We have used the instances of Ivancic et al. [14] for the input minimization problem. This data set is composed of 47 instances ranging from two to five types of boxes, and from 47 to 180 boxes. A single type of container is considered in these instances. The objective is to find the lowest number of containers needed to pack all the boxes. These instances have been solved by different methods, and we compare our results against the integer programming-based heuristic of Ivancic et al. [14], the heuristic box and container selection criteria of Bortfeldt [2], the column generation heuristic of Eley [10], the greedy heuristic of Lim and Zhang [16], and the column generation procedures of Ren et al. [21] and Zhu et al. [26]. Only the tree search heuristics with sequential and parallel strategy of Eley [9] and the column generation procedure of Zhu et al. [26] include the practical consideration of stability. Throughout the text, for simplicity, these approaches will be called IVA, BOR, ELY, LZG, CHE ZHU, ELS, ELP, and ZHU-ST, respectively.

The output maximization problem was tested on the 16 instances proposed by Mohanty et al. [18], which range from two to six types of boxes, with availability varying from 47 to 200 items, and with two and three different container types, with availability ranging from two to 15 containers. The objective is to maximize the value associated with the cargo. Again, different methods have been used to solve these instances. We compare our results against the sequential solution strategy using a column generation procedure of Mohanty et al. [18], the heuristics combined with the box and container selection criteria of Bortfeldt [2], the column generation heuristic combined with integer programming of Eley [10], the multi-start local search procedure of Takahara [23], and the tree search algorithm of Ren et al. [21]. These techniques will be called MOH, BOR, ELY, TAK, and REN, respectively.

4.2 Discretizations techniques

The mathematical formulations for the input minimization problem, defined by equations (7)–(10), and for the output maximization problem, given by expressions (12)–(15), were tested using the four techquines to generate the points that can be occupied by the front-left-bottom vertex of a box, as mentioned in Section 2. For input minimization problems, the container availability was defined as the upper bound $\mathcal{U}$ obtained solving the set cover problem presented in Section 3.1. Algorithm 1 shows the procedure used to solve the problems.

Algorithm 1 Strategy to solve input minimization problems

Require: Box dimensions, box availability and container dimensions

  Compute the lower bound $\mathcal{L}$

  Compute the upper bound $\mathcal{U}$

  if $\mathcal{L} < \mathcal{U}$ **then**

    Generate the model

    Set $e_j = 1$, $j \in \{1, \ldots, \mathcal{L}\}$

    Add symmetry breaking constraints $e_j \leq e_{j-1}$, $\forall j \in \{\mathcal{L}+1, \ldots, \mathcal{U}\}$

    Solve the model

  end if

  return Loading pattern, number of containers

Table 1 presents the number of containers required to load the boxes on the 47 instances of Ivancic et al. [14], the number of variables and constraints of each model, as well as the time required to obtain the solution with each of the techniques. As all the techniques tested were obtained, directly or indirectly, from the NP sets, this technique generated models with more variables and constraints. The time needed to solve the models obtained using the RNP approach was slightly lower in comparison with the ones with NP, even though the number of variables and constraints is the same in both approaches. Even though the RRP yielded models with the lowest number of constraints and variables, the MiM approach obtained similar results, with the guarantee of no loss of generality.

Tests performed with the 16 instances of Mohanty et al. [18] for the output maximization problem showed a similar outcome from those presented for the input minimization case, as can be seen in Table 2. Thus, for simplicity, we will only show the results obtained with the MiM in the following sections.

4.3 Results for input minimization problems

The formulations for the input minimization problem presented in this article were tested without and with the practical consideration of stability. For the tests considering load stability, the height was discretized using Normal Patterns to guarantee that no loss of generality occurs. To obtain the upper bound, we created the columns for the set cover problem first applying the G&R heuristic sequentially. Then, more columns were created using the adapted GRMod approach presented in Section 3.1, with parameters $\beta = 0$, $\beta = 0.5$ and $\beta = 1$. The boxes were loaded in two different strategies: filling the container through the length and through the width. As a stop criterion, we set a time limit of 15 seconds or a maximum of 100,000 executions in each direction for each parameter. After solving the set cover the optimality was proved comparing the upper and lower bound in 22 instances, as highlighted in Table 3. The upper bound $\mathcal{U}$ and the corresponding loading pattern for each instance were computed in an average time of 7.59 seconds.

Table 4 compares our results without the practical consideration of stability against the competing methods described in Section 4.1. Optimal solutions are highlighted in bold, and the best known solutions in italics. We improved two solutions and match all of the

Table 1: Comparison between the discretization techniques using the instances of Ivancic et al. [14]

#	NP				RNP				RRP				MIM			
	Containers	Variables	Constraints	Time (s)	Containers	Variables	Constraints	Time (s)	Containers	Variables	Constraints	Time (s)	Containers	Variables	Constraints	Time (s)
1	25	3725	3002	3.03	25	3725	3002	3.08	25	3725	3002	2.94	25	3725	3002	2.99
2	9	12730	3432	10.69	9	12730	3432	10.50	9	12730	3432	10.87	9	12730	3432	10.47
3	19	17917	1904	9.95	19	17917	1904	9.95	19	17917	1904	9.95	19	17917	1904	9.95
4	26	14326	1876	5.84	26	14326	1876	5.84	26	14326	1876	5.84	26	14326	1876	5.84
5	51	10506	2044	27.74	51	10506	2044	27.18	51	10506	2044	27.29	51	10506	2044	27.32
6	10	4730	603	6.42	10	4730	603	6.42	10	4730	603	6.42	10	4730	603	6.42
7	16	3856	579	4.43	16	3856	579	4.43	16	3856	579	4.43	16	3856	579	4.43
8	4	5556	579	8.26	4	5556	579	8.26	4	5556	579	8.26	4	5556	579	8.26
9	19	8037	2662	14.15	19	8037	2662	12.84	19	8037	2662	13.64	19	8037	2662	14.87
10	55	1485	552	2.56	55	1485	552	2.56	55	1485	552	2.56	55	1485	552	2.56
11	16	9760	4482	43.27	16	9760	4482	4.20	16	9760	4482	43.41	16	9760	4482	41.79
12	53	9699	2123	4.20	53	9699	2123	4.22	53	9699	2123	4.09	53	9699	2123	4.20
13	25	24325	3753	7.15	25	24325	3753	7.02	25	24325	3753	7.15	25	24325	3753	7.07
14	27	13878	2433	4.40	27	13878	2433	4.40	27	13878	2433	4.40	27	13878	2433	4.40
15	11	23903	4227	5.34	11	23903	4227	5.34	11	23903	4227	5.34	11	23903	4227	5.34
16	26	10842	3123	5.58	26	10842	3123	5.43	26	10842	3123	5.44	26	10842	3123	5.49
17	7	35665	5085	7.46	7	35665	5085	7.46	7	35665	5085	7.46	7	35665	5085	7.46
18	2	88562	12379	13.07	2	88562	12379	13.07	2	88562	12379	13.07	2	88562	12379	13.07
19	3	77991	13203	9.11	3	77991	13203	9.11	3	77991	13203	9.11	3	77991	13203	9.11
20	5	39905	10403	195.16	5	39905	10403	152.01	5	39905	10403	130.51	5	39905	10403	318.54
21	20	90960	30805	7080.26	20	90960	30805	7081.64	20	90960	30805	6979.15	20	90960	30805	7054.96
22	8	287416	39173	15.00	8	287416	39173	15.00	8	287416	39173	15.00	8	287416	39173	15.00
23	19	62244	27221	3246.39	19	62244	27221	3268.13	19	62244	27221	3239.20	19	62244	27221	3261.04
24	5	162145	19604	10.74	5	162145	19604	10.74	5	162145	19604	10.74	5	162145	19604	10.74
25	5	314405	29164	7200.00	5	314405	29164	7200.00	5	314405	29164	7200.00	5	314405	29164	7200.00
26	3	506889	38644	15.00	3	506889	38644	15.00	3	506889	38644	15.00	3	506889	38644	15.00
27	4	112940	20067	8.17	4	112940	20067	8.17	4	112940	20067	8.17	4	112940	20067	8.17
28	9	48753	15879	6.79	9	48753	15879	6.79	9	48753	15879	6.79	9	48753	15879	6.79
29	16	80160	25924	6030.58	16	80160	25924	5846.36	16	78480	19716	4316.05	16	78480	19716	4333.17
30	22	49082	22444	7200.00	22	49082	22444	7200.00	22	47344	15844	7200.00	22	47344	15844	7200.00
31	12	168987	36040	7200.00	12	168987	36040	7200.00	12	165841	28734	7200.00	12	165841	28734	7200.00
32	4	110452	16019	9.44	4	110452	16019	9.44	4	110452	16019	9.44	4	110452	16019	9.44
33	4	61028	12099	8.14	4	61028	12099	8.14	4	61028	12099	8.14	4	61028	12099	8.14
34	8	43416	10755	108.53	8	43416	10755	104.32	8	43416	10755	107.20	8	43416	10755	107.36
35	2	64834	12802	5.94	2	64834	12802	5.94	2	61234	10832	5.94	2	61234	10832	5.94
36	14	10528	7282	6.01	14	10528	7282	6.12	14	10136	6050	6.00	14	10136	6050	5.79
37	23	63319	11963	17.06	23	63319	11963	17.38	23	63319	11963	17.64	23	63319	11963	17.41
38	45	31995	8643	9.27	45	31995	8643	9.55	45	31995	8643	9.74	45	31995	8643	9.72
39	15	47745	8193	726.13	15	47745	8193	704.02	15	47745	8193	717.54	15	47745	8193	735.97
40	8	133432	16900	3660.31	8	133432	16900	3550.23	8	133432	16900	3647.12	8	133432	16900	3660.02
41	15	61755	11344	59.00	15	61755	11344	57.51	15	61755	11344	59.42	15	61755	11344	59.17
42	4	260532	27107	9.78	4	260532	27107	9.78	4	148244	14595	9.78	4	157540	17027	9.78
43	3	756147	71985	15.00	3	756147	71985	15.00	3	475659	38853	15.00	3	475659	38853	15.00
44	3	599652	62643	5987.74	3	599652	62643	5927.74	3	325540	28707	830.11	3	398116	35883	1255.09
45	3	1326255	113404	7200.00	3	1326255	113404	7200.00	3	1257063	93916	7200.00	3	1257063	93916	7200.00
46	2	1335726	109392	15.00	2	1335726	109392	15.00	2	1270058	91732	15.00	2	1270058	91732	15.00
47	3	675213	74254	15.00	3	675213	74254	15.00	3	620871	58036	15.00	3	620871	58036	15.00
Average		167732.09	20387.11	1196.87		167732.09	20387.11	1187.45		149292.81	17062.38	1046.50		151034.77	17266.81	1062.62

Table 2: Comparison between the discretization techniques using the instances of Mohanty et al. [18]

#	NP				RNP				RRP				MIM			
	Obj. Value	Variables	Constraints	Time (s)	Obj. Value	Variables	Constraints	Time (s)	Obj. Value	Variables	Constraints	Time (s)	Obj. Value	Variables	Constraints	Time (s)
1	9216.00	2840	928	17.71	9216.00	2840	928	18.26	9216.00	2840	928	18.05	9216.00	2840	928	18.05
2	85555.20	8485	1064	12.59	85555.20	8485	1064	13.73	85555.20	8485	1064	12.89	85555.20	8485	1064	12.89
3	53262.50	2340	279	1.52	53262.50	2340	279	1.35	53262.50	2340	279	1.83	53262.50	2340	279	1.23
4	2354752.00	2240	752	0.07	2354752.00	2240	752	0.07	2354752.00	2240	752	0.07	2354752.00	2240	752	0.07
5	583750.00	4488	753	1.41	583750.00	4488	753	0.70	583750.00	4488	753	0.50	583750.00	4488	753	0.50
6	142464.00	16612	2823	7200.00	142464.00	16612	2823	7200.00	142464.00	16612	2823	7200.00	142464.00	16612	2823	7200.00
7	17664.00	33976	6483	7200.00	17664.00	33976	6483	7200.00	17664.00	33976	6483	7200.00	17664.00	33976	6483	7200.00
8	71972.40	204915	25310	7200.00	71972.40	204915	25310	7200.00	71972.40	199821	21118	7200.00	71972.40	199821	21118	7200.00
9	114228.00	264270	22636	7200.00	114228.00	264270	22636	7200.00	114228.00	264270	22636	7200.00	114228.00	264270	22636	7200.00
10	15360.00	5392	1283	1.75	15360.00	5392	1283	1.82	15360.00	5392	1283	0.99	15360.00	5392	1283	0.97
11	54761.00	25246	7036	1174.91	54761.00	25246	7036	1189.56	54761.00	24715	5398	312.78	54761.00	24715	5398	336.04
12	24393.60	48294	8375	7200.00	24393.60	48294	8375	7200.00	24393.60	48294	8375	7200.00	24393.60	48294	8375	7200.00
13	36556.80	66334	13842	414.42	36556.80	66334	13842	428.50	36556.80	62678	11696	377.10	36556.80	62678	11696	461.63
14	68723.20	13288	2519	7200.00	68723.20	13288	2519	7200.00	68723.20	13288	2519	7150.31	68723.20	13288	2519	6547.81
15	40807.80	41588	5740	26.89	40807.80	41588	5740	25.65	40807.80	41588	5740	34.36	40807.80	41588	5740	27.29
16	498426.00	323197	35677	7200.00	481434.00	323197	35677	7200.00	629754.00	154874	14633	7200.00	632274.00	179988	17160	7200.00
Average		66469.06	8468.75	3253.20		66469.06	8468.75	3254.97		55368.81	6655.00	3194.30		56938.43	6812.93	3162.90

BKS for this 47 instance set. Moreover, we prove optimality for 43 instances, the highest number in literature. We obtained outstanding results, optimally solving 91.48% of the instances, against 53.19% from Eley [10]. Comparing the Tables 3 and 4, besides the 22 instances which optimality was proved early in Table 3, the set cover approach found the optimal solution in 18 cases, later proved by solving the models. Overall, the sum of containers needed to pack the boxes is 688, three fewer than the approach of Zhu et al. [26], currently the best in CLP literature.

The stability coefficient was defined as $\alpha = 1$ for tests with this constraint, i.e., the bottom side of each box must be fully supported by other boxes or by the floor of the container. This value is commonly found in the literature, being used, for example, by Eley [9] and Junqueira et al. [15]. Table 5 shows the results considering the stability constraint. We compare our results against the approaches of Eley [9] and Zhu et al. [26]. Out of the 47 instances, we proved optimality for 37 (highlighted in bold). Our results were equal to the BKS in 43 instances and we improved the BKS for two cases (BKS in italic), requiring one fewer container in each instance to load the boxes.

Figures 3 and 4 show the loading pattern obtained for instance 8 without and with the stability consideration, respectively.

Table 3: Bounds for the instances of Ivancic et al. [14]

#	Lower bound	Upper bound	Time (s)	#	Lower bound	Upper bound	Time (s)
1	19	25	2.55	25	4	5	5.82
2	7	10	5.96	26	**3**	**3**	15
3	**19**	**19**	9.95	27	**4**	**4**	8.17
4	**26**	**26**	5.84	28	**9**	**9**	6.79
5	46	51	6.43	29	15	16	7.09
6	**10**	**10**	6.42	30	18	22	5.24
7	**16**	**16**	4.43	31	11	13	7.01
8	**4**	**4**	8.26	32	**4**	**4**	9.44
9	16	19	2.42	33	**4**	**4**	8.14
10	37	55	2.54	34	7	8	6.16
11	14	16	3.06	35	**2**	**2**	5.94
12	45	53	3.63	36	10	14	3.68
13	20	25	4.57	37	12	23	7.92
14	**27**	**27**	4.4	38	25	45	5.39
15	**11**	**11**	5.34	39	12	15	4.77
16	21	26	4.53	40	7	8	12.93
17	**7**	**7**	7.46	41	14	15	5.99
18	**2**	**2**	13.07	42	**4**	**4**	9.78
19	**3**	**3**	9.11	43	**3**	**3**	15
20	4	5	4.61	44	3	4	6.37
21	17	20	8.08	45	2	3	15
22	**8**	**8**	15	46	**2**	**2**	15
23	17	21	6.62	47	**3**	**3**	15.00
24	**5**	**5**	10.74	Sum/Avg	579	693	7.59

Bold indicates proven optimal

Table 4: Results for the instances of Ivancic et al. [14]

#	IVA	BOR	ELY	LZG	CHE	ZHU	This paper	
							Containers	Deviation to the BKS
1	26	25	25	25	25	25	***25***	0
2	11	10	10	10	10	10	***9***	-1
3	20	20	20	19	19	19	***19***	0
4	27	28	26	26	26	26	***26***	0
5	65	51	51	51	51	51	***51***	0
6	10	10	10	10	10	10	***10***	0
7	16	16	16	16	16	16	***16***	0
8	5	4	4	4	4	4	***4***	0
9	19	19	19	19	19	19	***19***	0
10	55	55	55	55	55	55	***55***	0
11	18	18	17	16	16	16	***16***	0
12	55	53	53	53	53	53	***53***	0
13	27	25	25	25	25	25	***25***	0
14	28	28	27	27	27	27	***27***	0
15	11	11	11	11	11	11	***11***	0
16	34	26	26	26	26	26	***26***	0
17	8	7	7	7	7	7	***7***	0
18	3	2	2	2	2	2	***2***	0
19	3	3	3	3	3	3	***3***	0
20	5	5	5	5	5	5	***5***	0
21	24	21	20	20	20	20	***20***	0
22	10	9	8	9	8	8	***8***	0
23	21	20	20	20	19	19	***19***	0
24	6	6	6	5	5	5	***5***	0
25	6	5	5	5	5	5	*5*	0
26	3	3	3	3	3	3	***3***	0
27	5	5	5	5	5	4	***4***	0
28	10	10	10	9	10	10	***9***	0

Continued on next page

Table 4: Results for the instances of Ivancic et al. [14]

#	IVA	BOR	ELY	LZG	CHE	ZHU	This paper	
							Containers	Deviation to the BKS
29	18	17	17	17	17	17	***16***	-1
30	24	22	22	22	22	22	*22*	0
31	13	13	13	12	12	12	*12*	0
32	5	4	4	4	4	4	***4***	0
33	5	5	5	4	4	4	***4***	0
34	9	8	8	8	8	8	***8***	0
35	3	2	2	2	2	2	***2***	0
36	18	14	14	14	14	14	***14***	0
37	26	23	23	23	23	23	***23***	0
38	50	45	45	45	45	45	***45***	0
39	16	15	15	15	15	15	***15***	0
40	9	9	8	9	8	8	***8***	0
41	16	15	15	15	15	15	***15***	0
42	4	4	4	4	4	4	***4***	0
43	3	3	3	3	3	3	***3***	0
44	4	3	4	3	3	3	***3***	0
45	3	3	3	3	3	3	*3*	0
46	2	2	2	2	2	2	***2***	0
47	4	3	3	3	3	3	***3***	0
Sum	763	705	699	694	692	691	**688**	

Italic indicates BKS

Bold indicates proven optimal

Table 5: Results for the instances of Ivancic et al. [14] with load stability

#	ELS	ELP	ZHU-ST	This paper	
				Containers	Deviation to the BKS
1	27	26	25	***25***	0

Continued on next page

Table 5: Results for the instances of Ivancic et al. [14] with load stability

#	ELS	ELP	ZHU-ST	This paper	
				Containers	Deviation to the BKS
2	11	10	10	*10*	0
3	21	22	19	***19***	0
4	29	30	26	***26***	0
5	55	51	51	***51***	0
6	10	10	10	***10***	0
7	16	16	16	***16***	0
8	4	4	4	***4***	0
9	19	19	19	***19***	0
10	55	55	55	***55***	0
11	17	18	17	***16***	-1
12	53	53	53	***53***	0
13	25	25	25	***25***	0
14	27	27	27	***27***	0
15	12	12	11	***11***	0
16	28	26	26	***26***	0
17	8	7	7	***7***	0
18	2	2	2	***2***	0
19	3	3	3	***3***	0
20	5	5	5	***5***	0
21	24	26	20	*20*	0
22	9	9	8	***8***	0
23	21	21	20	21	1
24	6	6	5	***5***	0
25	6	5	5	*5*	0
26	3	3	3	***3***	0
27	5	5	4	***4***	0
28	11	10	10	***9***	-1
29	18	18	17	*17*	0

Continued on next page

Table 5: Results for the instances of Ivancic et al. [14] with load stability

#	ELS	ELP	ZHU-ST	This paper	
				Containers	Deviation to the BKS
30	22	23	22	*22*	0
31	13	14	12	13	1
32	4	4	4	***4***	0
33	5	5	4	***4***	0
34	8	9	8	***8***	0
35	2	2	2	***2***	0
36	18	14	14	***14***	0
37	26	23	23	***23***	0
38	46	45	45	***45***	0
39	15	15	15	*15*	0
40	9	9	8	*8*	0
41	16	15	15	***15***	0
42	4	4	4	***4***	0
43	3	3	3	***3***	0
44	4	4	3	***3***	0
45	3	3	3	*3*	0
46	2	2	2	***2***	0
47	3	3	3	***3***	0
Sum	733	721	693	693	

Italic indicates BKS

Bold indicates proven optimal

4.4 Results for output maximization problems

To evaluate the formulation for the maximization problem, we considered four different situations: first, we only assessed the formulation given by (12)–(15). Then, we considered the constraints for the separation of boxes, considering the stability of the cargo, and finally with both practical considerations. To ensure no loss of generality, in the tests

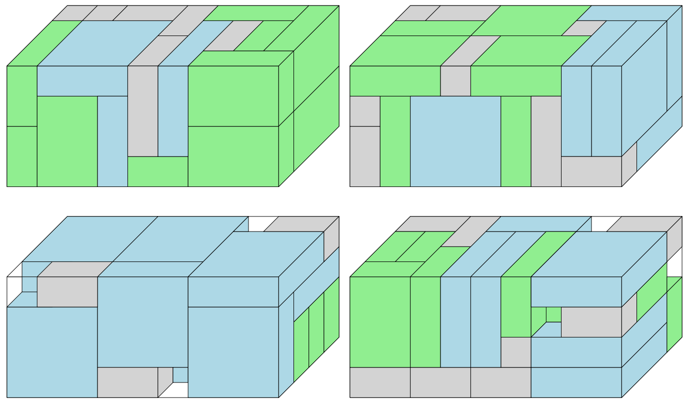

Figure 3: Loading patterns for instance 8 of Ivancic et al. [14].

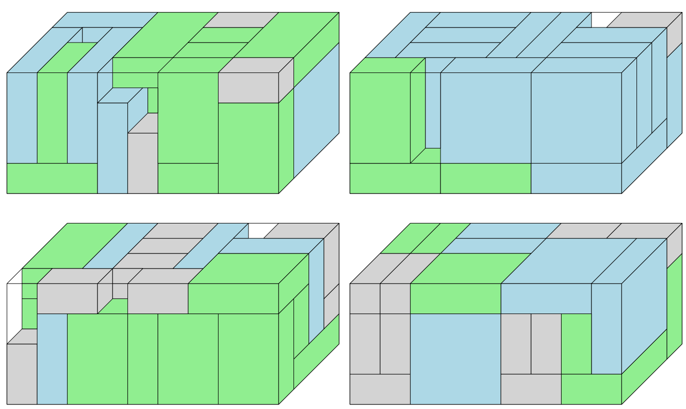

Figure 4: Loading patterns for instance 8 of Ivancic et al. [14] with load stability.

with stability consideration the height was discretized using Normal Patterns due to this requirement.

Table 6 compares the results obtained by our approach against the ones mentioned in Section 4.1. In 15 out of the 16 instances, our approach yielded the BKS, improving the existing one for 10 instances. Those are highlighted in italics. Additionally, 10 instances, highlighted in bold, were solved optimally.

Table 6: Results for the test sets of Mohanty et al. [18]

#	MOH	BOR	ELY	TAK	REN	Proposed approach		
						Obj. Value	Time (s)	Gap (%)
1	8640.00	8640.00	8640.00	8640.00	8640.00	***9216.00***	18.05	0.00
2	83494.40	85120.00	85376.00	84224.00	85376.00	***85555.20***	12.89	0.00
3	53262.50	53262.50	53262.50	52350.00	53262.50	***53262.50***	1.23	0.00
4	2333440.00	2333440.00	2307840.00	2333440.00	2333440.00	***2354752.00***	0.07	0.00
5	495500.00	581250.00	583750.00	579250.00	579250.00	***583750.00***	0.50	0.00
6	138240.00	139584.00	141216.00	137952.00	139968.00	*142464.00*	7200.00	0.47
7	16668.00	17409.00	17004.00	17262.00	17226.00	*17664.00*	7200.00	1.02
8	65741.00	68645.60	69121.20	69747.20	71236.40	*71972.40*	7200.00	0.40
9	119772.00	128952.00	133632.00	128556.00	130860.00	114228.00	7200.00	-
10	15360.00	15360.00	15360.00	15360.00	15360.00	***15360.00***	0.99	0.00
11	49995.00	53202.80	52873.60	53202.80	53202.80	***54761.00***	312.78	0.00
12	23529.00	24235.20	23673.00	23990.40	23990.40	*24393.60*	7200.00	1.45
13	36556.80	36556.80	36556.80	36556.80	36556.80	***36556.80***	377.11	0.00
14	56492.80	65316.80	68723.20	68723.20	68723.20	***68723.20***	7150.31	0.00
15	37558.80	39727.20	39382.20	40590.00	40590.00	***40807.80***	34.36	0.00
16	556458.00	595770.00	591535.00	571290.00	603000.00	*632274.00*	7200.00	2.40

Bold indicates optimal solution

Italic indicates BKS

- indicates that the solver did not provide a dual bound

For the separation of boxes, we followed Eley [10] imposing that boxes of types 1 and 2 be separated. Our results improve the solution of Eley [10] in 11 instances, and we prove optimality for 9 instances. Detailed solutions are reported in Table 7.

Table 7: Results for the test sets of Mohanty et al. [18] with separation of boxes

#	ESI	Proposed approach		
		Obj. Value	Time (s)	Gap (%)
1	5120.00	***7680.00***	4.31	0.00
2	85376.00	***85555.20***	2.51	0.00
3	53262.50	***53262.50***	4.08	0.00
4	1354752.00	***1354752.00***	0.10	0.00
5	536250.00	***538750.00***	4.15	0.00
6	139968.00	*140448.00*	7200.00	0.34
7	16707.00	*17664.00*	7200.00	0.78
8	69121.20	*71972.40*	7200.00	0.40
9	128088.00	102672.00	7200.00	-
10	15360.00	***15360.00***	2.02	0.00
11	52873,60	***54761.00***	433.38	0.00
12	22730.40	*23745.60*	7200.00	4.15
13	34022.40	*34022.40*	7200.00	7.45
14	66995.20	***66995.20***	3524.10	0.00
15	39382.20	***40807.80***	29.96	0.00
16	568482.00	*612546.00*	7200.00	5.62

Bold indicates optimal solution

Italic indicates BKS

- indicates that the solver did not provide a dual bound

The stability constant was again defined with $\alpha = 1$, i.e., requiring 100% of support to the bottom of the boxes. Although the load stability requirement significantly increases the complexity of the problem, we obtained an optimal solution for eight instances. When the practical considerations of load stability and separation of boxes are jointly considered, the objective value is typically much worse due to the more constrained nature of the problem. Nevertheless, the proposed formulation obtained an optimal solution for seven instances. Results for these scenarios are shown in Table 8.

Table 8: Results for the test sets of Mohanty et al. [18]

#	With load stability			With load stability and separation of boxes		
	Obj. Value	Time (s)	Gap (%)	Obj. Value	Time (s)	Gap (%)
1	**8640.00**	3169.20	0.00	**6720.00**	55.19	0.00
2	85376.00	7200.00	0.21	85376.00	7200.00	0.19
3	**53262.50**	20.88	0.00	**53262.50**	17.31	0.00
4	**1354752.00**	1.59	0.00	**1354752.00**	0.96	0.00
5	**583750.00**	8.60	0.00	**538750.00**	22.61	0.00
6	142464.00	7200.00	0.67	139968.00	7200.00	2.40
7	16896.00	7200.00	6.20	17037.00	7200.00	5.14
8	40556.20	7200.00	-	34255.60	7200.00	-
9	*	7200.00	-	*	7200.00	-
10	**15360.00**	4.63	0.00	**15360.00**	4.96	0.00
11	**54761.00**	2311.70	0.00	**54761.00**	4732.53	0.00
12	18504.00	7200.00	-	16992.00	7200.00	-
13	**36556.80**	466.42	0.00	34022.40	7200.00	7.45
14	68723.20	7200.00	0.88	66995.20	7200.00	1.58
15	**40807.80**	1500.70	0.00	**40807.80**	2008.30	0.00
16	481434.00	7200.00	-	304200.00	7200.00	-

Bold indicates optimal solution

- indicates that the solver did not provide a dual bound

* indicates that the solver did not provide a solution in the time set for the test

Figures 5–8 show the loading pattern obtained for instance 12 in each of the evaluated cases. All boxes have their bases fully supported by other items or by the floor of the container when the load stability is considered, and in cases considering the separation of boxes, items of type 1 and 2 (in gray and blue, respectively) were allocated to different containers.

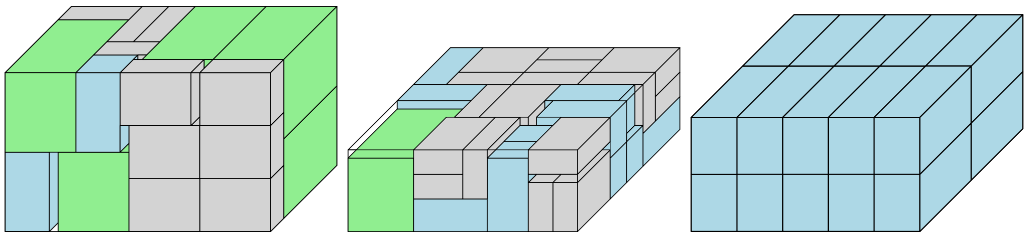

Figure 5: Loading patterns for instance 12 of Mohanty et al. [18].

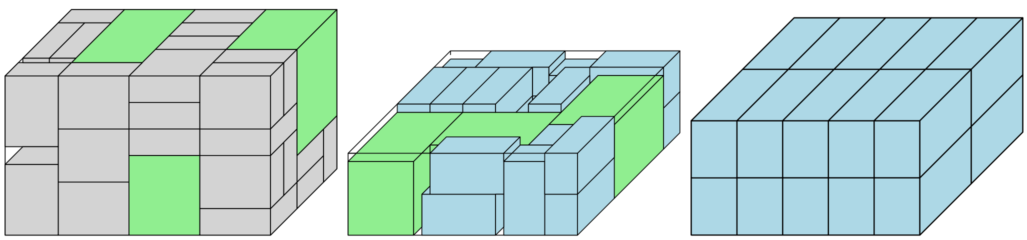

Figure 6: Loading patterns for instance 12 of Mohanty et al. [18] with separation of boxes.

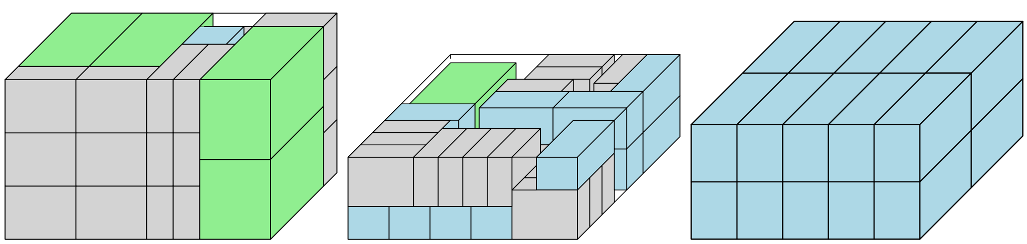

Figure 7: Loading patterns for instance 12 of Mohanty et al. [18] with load stability.

5 Conclusion

This paper presented mathematical formulations for several classes of multiple container loading problems. We have adapted and tested four discretization techniques to enumer-

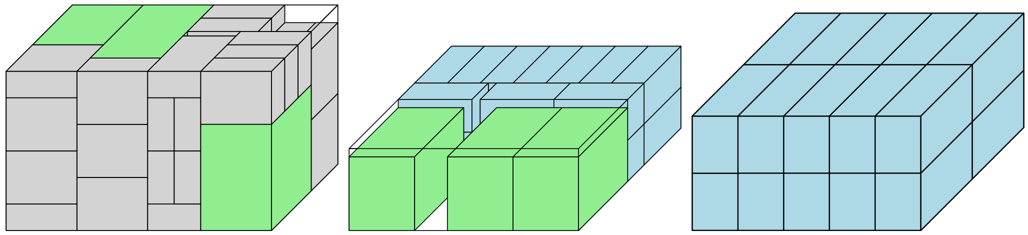

Figure 8: Loading patterns for instance 12 of Mohanty et al. [18] with load stability and separation of boxes.

ate the possible positions of a box inside a container, and also presented an enhanced version of the classic heuristic of George and Robinson [11] to obtain an upper bound for input minimization problems. Besides, we have also proposed mathematical formulations to the practical considerations of separation of boxes and stability, allowing the boxes to assume up to six orientations in our approaches. Computational experiments using well known data sets in the literatures were carried out and in many cases new optimal or improved BKS were obtained. Specifically, we have improved two solutions for the input minimization and matched 45 BKS out of the 47 instances available; for output maximization, for the 16 instances available we have improved the BKS in ten instances and matched five BKS.

The formulations presented in this article can be used to create new techniques combining exact methods with heuristic or metaheuristics strategies, to achieve good solutions in shorter computational time. Also, new formulations to obtain tighter bounds can be devised to assess the quality of the solutions for the problems.

Mathematical formulations for other practical considerations can be taken into account, in order to address more realistic problems. Finally, the formulations presented in this article can be combined with routing algorithms, formulating approaches to the capacitated vehicle routing problem with three dimensional load requirements.

References

References

[1] E.E. Bischoff and M.S.W. Ratcliff. Issues in the development of approaches to container loading. *Omega*, 23(4):377 – 390, 1995.

[2] A. Bortfeldt. Eine heuristik für multiple containerladeprobleme. *OR-Spektrum*, 22 (2):239–261, 2000.

[3] A. Bortfeldt and G. Wäscher. Constraints in container loading – a state-of-the-art review. *European Journal of Operational Research*, 229(1):1 – 20, 2013.

[4] M. A. Boschetti, A. Mingozzi, and E. Hadjiconstantinou. New upper bounds for the two-dimensional orthogonal non-guillotine cutting stock problem. *IMA Journal of Management Mathematics*, 13(2):95–119, 2002.

[5] C. H. Che, W. Huang, A. Lim, and W. Zhu. The multiple container loading cost minimization problem. *European Journal of Operational Research*, 214(3):501 – 511, 2011.

[6] N. Christofides and C. Whitlock. An algorithm for two-dimensional cutting problems. *Operations Research*, 25(1):30–44, 1977.

[7] J.-F. Côté and M. Iori. The meet-in-the-middle principle for cutting and packing problems. Technical Report CIRRELT-2016-28, Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport - CIRRELT, Université Laval, Québec, Canada, June 2016.

[8] T. A. de Queiroz, F. K. Miyazawa, Y. Wakabayashi, and E. C. Xavier. Algorithms for 3D guillotine cutting problems: Unbounded knapsack, cutting stock and strip packing. *Computers & Operations Research*, 39(2):200 – 212, 2012.

[9] M. Eley. Solving container loading problems by block arrangement. *European Journal of Operational Research*, 141(2):393 – 409, 2002.

[10] M. Eley. A bottleneck assignment approach to the multiple container loading problem. *OR Spectrum*, 25(1):45–60, 2003.

[11] J.A. George and D.F. Robinson. A heuristic for packing boxes into a container. *Computers & Operations Research*, 7(3):147 – 156, 1980.

[12] M. Grunewald, T. Volling, C. Müller, and T. S. Spengler. Multi-item single-source ordering with detailed consideration of transportation capacities. *Journal of Business Economics*, forthcoming, Mar 2018.

[13] J. C. Herz. Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development*, 16(5):462–469, 1972.

[14] N. Ivancic, K. Mathur, and B. Mohanty. An integer programming based heuristic approach to the three-dimensional packing problem. *Journal of Manufacturing and Operations Management*, 2:268 – 298, 1989.

[15] L. Junqueira, R. Morabito, and D. S. Yamashita. Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers & Operations Research*, 39(1):74 – 85, 2012.

[16] A. Lim and X. Zhang. The container loading problem. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, pages 913–917, New York, NY, USA, 2005. ACM.

[17] S. Martello and P. Toth. Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics*, 28(1):59 – 70, 1990.

[18] B. B. Mohanty, K. Mathur, and N. J. Ivancic. Value considerations in three-dimensional packing — a heuristic procedure using the fractional knapsack problem. *European Journal of Operational Research*, 74(1):143 – 151, 1994.

[19] A. Moura and J. F. Oliveira. A GRASP approach to the container-loading problem. *IEEE Intelligent Systems*, 20(4):50–57, 2005.

[20] A. G. Ramos, E. Silva, and J. F. Oliveira. A new load balance methodology for container loading problem in road transportation. *European Journal of Operational Research*, 266(3):1140 – 1152, 2018.

[21] J. Ren, Y. Tian, and T. Sawaragi. A priority-considering approach for the multiple container loading problem. *International Journal of Metaheuristics*, 1(4):298–316, 2011.

[22] G. Scheithauer and J. Terno. The G4-heuristic for the pallet loading problem. *The Journal of the Operational Research Society*, 47(4):511–522, 1996.

[23] S. Takahara. A multi-start local search approach to the multiple container loading problem. In W. Bednorz, editor, *Greedy Algorithms*, chapter 4, pages 55–68. InTech, Rijeka, 2008.

[24] J. Terno, R. Lindemann, and G. Scheithauer. *Zuschnittprobleme und ihre praktische Lösung*. Mathematik für Ingenieure. Fachbuchverl, 1987.

[25] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109 – 1130, 2007.

[26] W. Zhu, W. Huang, and A. Lim. A prototype column generation strategy for the multiple container loading problem. *European Journal of Operational Research*, 223 (1):27 – 39, 2012.