Ann Oper Res (2012) 199:51–75
DOI 10.1007/s10479-011-0942-z

MIP-based approaches for the container loading problem with multi-drop constraints

Leonardo Junqueira · Reinaldo Morabito · Denise Sato Yamashita

Published online: 28 August 2011

Abstract In this paper, we present approaches based on a mixed integer linear programming model (MIP) for the problem of packing rectangular boxes into a container or truck, considering multi-drop constraints. We assume that the delivery route of the container is already known in advance and that the volume of the cargo is less than or equal to the container volume. Considering the sequence that the boxes should be unloaded, the aim is to avoid additional handling when each drop-off point of the route is reached, as well as ensuring that the boxes do not overlap each other and the cargo loading is stable. Computational tests with the proposed model and the approaches were performed with randomly generated instances and instances from the literature using an optimization solver embedded into a modeling language. The results validate the model and the approaches, but indicate that they are able to handle only problems of a moderate size. However, the model and the approaches can be useful to motivate future research to solve larger problems, as well as to solve more general problems considering integrated vehicle routing and container loading problems.

Keywords Three-dimensional container loading · Multi-drop constraints · Cutting and packing problems · Combinatorial optimization · Mathematical modeling

1 Introduction

Three-dimensional container loading problems satisfy two basic constraints: (i) the boxes should be completely packed inside the containers, and (ii) they should not overlap each other. A number of studies dealing with container loading problems are found in the literature considering these constraints, such as in George and Robinson (1980), Han et al.

L. Junqueira · R. Morabito (✉) · D. Sato Yamashita
Departamento de Engenharia de Produção, Universidade Federal de São Carlos, São Carlos, Brazil
e-mail: morabito@ufscar.br

L. Junqueira
e-mail: leo_junqueira@yahoo.com

D. Sato Yamashita
e-mail: denisesy@dep.ufscar.br

(1989), Bischoff and Marriott (1990), Haessler and Talbot (1990) and Dowsland (1991). Other work appears in, e.g., Morabito and Arenales (1994), Bischoff and Ratcliff (1995), Miyazawa and Wakabayashi (1999), Eley (2002), Lins et al. (2002), Silva et al. (2003) and Parreño et al. (2010). Additional practical constraints have also been considered when dealing with these problems. In Bischoff and Ratcliff (1995) twelve practical considerations that could be taken into account concerning modeling and solving more realistic container loading problems were presented. Constraints such as cargo stability, load bearing strength of the cargo (including fragility), multi-dropping, weight limit, weight distribution inside the container, among others, are often common and important in practice.

In this paper, we present MIP-based approaches to solve the problem of loading rectangular boxes into a single rectangular container considering multi-drop constraints. Besides the aforementioned constraints (i) and (ii), these approaches can be easily adapted to cope with vertical and/or horizontal stability of the cargo, load bearing strength and fragility of the boxes and weight limit of the cargo. Multi-drop constraints consider that boxes to be delivered to the same customers (destinations) should be placed close to each other inside the container (or truck, as it is more common in practice), and the loading patterns of boxes must take into account the delivery route of the vehicle and the sequence in which the boxes are unloaded on this route to avoid additional box handling when each drop-off point is reached. To the best of our knowledge, there is not much work in the literature concerning container loading problems with multi-dropping.

Multi-drop constraints can be dealt with in two ways. In the first, we assume that the delivery route of the container or truck is already known in advance, and we are interested in finding the best arrangement of the boxes inside the container. In the second, we assume that the route is still not established, which leads to a combined approach of determining the delivery route of the container and arranging the boxes inside the container. In this paper, we are particularly interested in the first situation and the aim is to determine the best loading pattern, ensuring that constraints (i) and (ii) above are met and taking into account the sequence in which the boxes must be unloaded from the container, without additional handling requirements when each drop-off point is reached. In case there is not enough space to pack all boxes inside the container (i.e., constraint (i) cannot be met), the best loading pattern is the one that maximizes the total volume (or value) of the boxes packed. Some related studies can be found in, e.g., Bischoff and Ratcliff (1995), Scheithauer et al. (1996), Lai et al. (1998), Terno et al. (2000), Jin et al. (2004), Moura and Oliveira (2005), Lin et al. (2006), Iori et al. (2007), Christensen and Rousøe (2009), Moura and Bortfeldt (2009) and Moura and Oliveira (2009). The second situation considering the integrated vehicle routing and container loading problem is beyond the scope of this study and it is an interesting topic for future research. Some examples are found in Gendreau et al. (2006), Moura and Oliveira (2009), Tarantilis et al. (2009) and Fuellerer et al. (2010).

This work is organized as follows. In Sect. 2, we describe in more details the container loading with multi-drop constraints and we present a MIP formulation for the problem, considering as before that the route of the container or truck has already been established. In this model, we also consider additional constraints, such as the vertical stability of the cargo. In Sect. 3, we present two solution approaches based on the model in Sect. 2. Randomly generated instances and instances from the literature are used to evaluate the performance of these approaches, coded in the modeling language GAMS and solved by the CPLEX solver. In Sect. 4, the results of these computational tests are presented and analyzed. Finally, in Sect. 5, we present concluding remarks and some perspectives for future research.

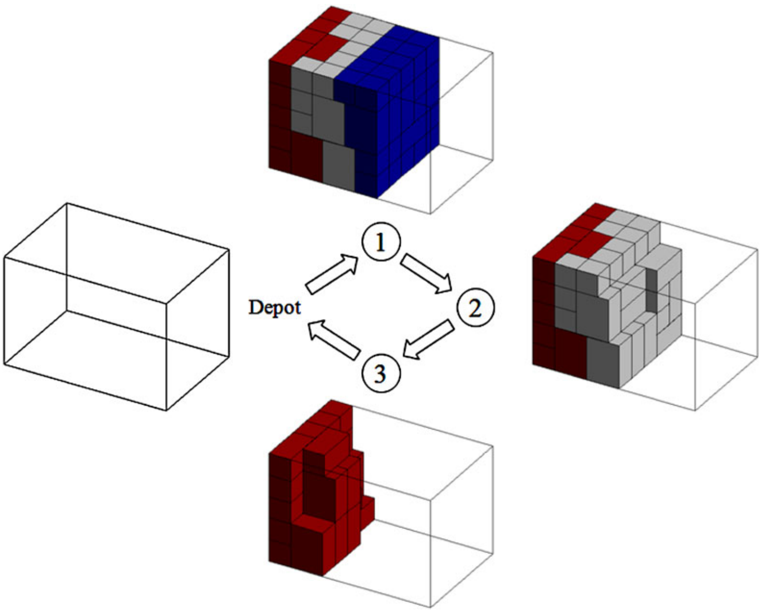


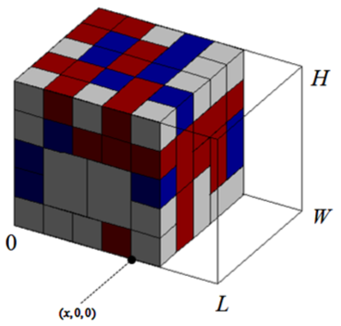
Fig. 1 A container is unloaded in three destinations

2 Problem description and mathematical modeling

As mentioned before, the multi-drop constraints address situations where a container (or truck) is packed with boxes containing goods ordered by different customers (destinations), which are spread through a region. The container leaves the depot (where it is packed), and visits each destination in a predefined sequence, delivering the boxes containing the goods ordered by each customer. After delivering all the goods, the empty container returns to the depot. The question that arises is how to plan the loading of the container so as to consider (as much as possible) the sequence in which the boxes must be unloaded, in order to avoid that an additional amount of time is spent unloading and reloading boxes of the remaining destinations. Figure 1 shows a loaded container that leaves a depot and is unloaded in three destinations, before returning empty to the depot (i.e., the route depot-1-2-3-depot).

We observe that this problem can be conversely seen as the problem of an empty pick-up container (or truck) that leaves the depot and, in each destination of its route, the boxes are loaded inside it without having to unload the boxes already packed in the earlier destinations. Note that the pick-up route is in the inverse sequence of the delivery route (i.e., the route depot-3-2-1-depot). Figure 2 shows a container packed with boxes of three different destinations (depicted by the different colors of the boxes). The boxes must be unloaded according to the sequence already shown in Fig. 1. Note that if the boxes were packed as shown in Fig. 2 (left), unnecessary additional handling would probably be incurred when reaching each drop-off point, since some boxes would need to be unloaded and later reloaded. In this case, a simple rearrangement of these boxes inside the container, as shown in Fig. 2 (right), could avoid wasting time when unloading the container at a destination.

Let us now consider a container of dimensions (L, W, H) with a delivery route with n destinations. For each destination k ($k = 1, \dots, n$), there are b_{ik} boxes (of a total of b_i boxes of type $i, i = 1, \dots, m$, in all destinations), with length l_i, width w_i and height h_i, which



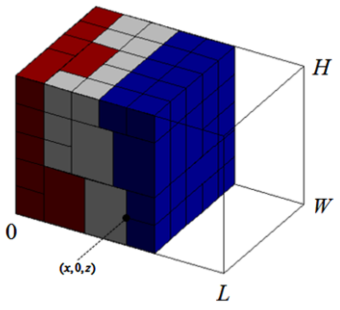


Fig. 2 Loading patterns with and without additional handling, respectively

must be loaded inside the container (we may have $b_{ik} = 0$ for some i and k). We assume that the total volume of the boxes ($\sum_{i=1}^{m}\sum_{k=1}^{n} l_i \cdot w_i \cdot h_i \cdot b_{ik}$) is less than or equal to the container volume ($L \cdot W \cdot H$). Furthermore, we assume that the dimensions of the boxes are integer, that the boxes can only be placed orthogonally into the container (i.e., the edges of a box are either parallel or perpendicular to the axes of the container), and that their orientation is fixed (i.e., the boxes cannot rotate). This last assumption can be easily relaxed in the models presented below and here it is considered only to simplify the presentation of the formulations. We note that $k = 1$ refers to the boxes that are loaded first and unloaded last, and $k = n$ refers to the boxes that are loaded last and unloaded first.

A Cartesian coordinate system is adopted with its origin in the container's front-left-bottom corner, and let (x, y, z) be the possible coordinates where the front-left-bottom corner of a box can be placed (see Fig. 2). These possible positions along axes x, y and z of the container belong to the sets: $X = \{0, 1, 2, \dots, L - \min_i(l_i)\}$, $Y = \{0, 1, 2, \dots, W - \min_i(w_i)\}$ and $Z = \{0, 1, 2, \dots, H - \min_i(h_i)\}$, respectively. Let $X_i = \{x \in X | 0 \le x \le L - l_i\}$, $Y_i = \{y \in Y | 0 \le y \le W - w_i\}$ and $Z_i = \{z \in Z | 0 \le z \le H - h_i\}$, $i = 1, \dots, m$, be defined as subsets of X, Y and Z, respectively.

We also define $\delta_{ik} \in [0, L]$ as a parameter relative to the reach of the worker (door's-man) tasked to manually arrange the boxes of destination k into the container, in terms of units of the container length (note that this parameter need not be equal for all k, but in some cases we may have $\delta_{ik} = \delta_i$ for all k, or even $\delta_{ik} = \delta$ for all k and i, where δ may be related to the arm's reach of the worker or to a forklift truck). When the boxes of a certain destination are already arranged into the container, the boxes of the next destination can be arranged taking advantage of possible empty spaces left behind by the boxes of the previous destinations. In other words, parameter δ_{ik} shows how many units of length beyond the "border" between boxes of consecutive destinations the worker is allowed to surpass in order to arrange the boxes of the coming destinations. The "border" is a plane (or virtual wall) of type $(x, 0, 0)$ that is defined after all boxes of a destination are packed inside the container. Figure 3 (left) shows the border (scratched) left behind by the boxes of a previous destination $(k - 1)$. From this border on, the worker is allowed to overstep up to δ_{ik} units of the container length in order to arrange the boxes of the coming destination (k). The boxes of destination k, in turn, once arranged, leave behind a new border, up to $\delta_{i(k+1)}$ units beyond which the boxes of the coming destination $(k + 1)$ could be arranged (Fig. 3, right).

The use of this parameter is particularly important in practice in order to preserve the integrity of the cargo. It prevents the worker from relying on or even stepping on the boxes of previous destinations already arranged, in order to put or remove some boxes of the coming destination, which could damage the products due to deformation of the boxes containing

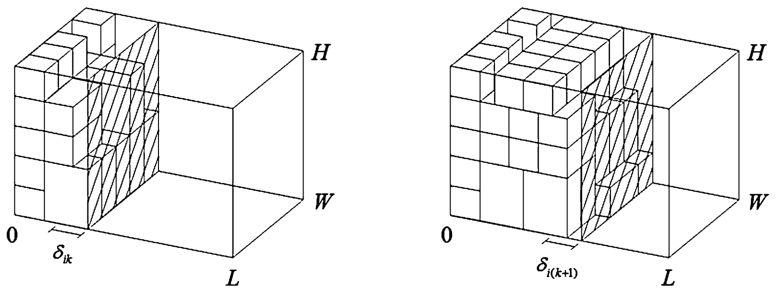


Fig. 3 Maximum reach of the worker

them. As aforementioned, this parameter can also represent the arm's reach of the worker, or even a piece of equipment used to load/unload the boxes, for instance, a forklift truck.

The decision variables a_{ikxyz}, $i = 1, \dots, m$, $k = 1, \dots, n$, $x \in X_i$, $y \in Y_i$, $z \in Z_i$, of the model are defined as:

$$a_{ikxyz} = \begin{cases} 1, & \text{if a box of type } i \text{ from destination } k \text{ is placed with its front-left-bottom} \\ & \text{corner at position } (x, y, z), \\ & \text{so that } 0 \le x \le L - l_i,\ 0 \le y \le W - w_i \text{ and } 0 \le z \le H - h_i; \\ 0, & \text{otherwise.} \end{cases}$$

Let us also define L'_k as the necessary length to load all boxes of destinations $1, 2, \dots, k$, and M as a sufficiently large number. Note that this variable defines the "border" aforementioned. Assuming that the container length L is sufficiently large to pack all boxes of all destinations, the problem of loading boxes inside the container with multi-dropping, and with additional considerations of cargo vertical stability, can be written as a direct extension of the 0–1 integer linear programming model recently presented in Junqueira et al. (2012):

$$\min \quad L'_n \tag{1}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{n}\sum_{\{x\in X_i \mid x'-l_i+1\le x\le x'\}}\ \sum_{\{y\in Y_i \mid y'-w_i+1\le y\le y'\}}\ \sum_{\{z\in Z_i \mid z'-h_i+1\le z\le z'\}} a_{ikxyz} \le 1$$
$$x' \in X,\ y' \in Y,\ z' \in Z \tag{2}$$

$$\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} a_{ikxyz} = b_{ik}, \quad i = 1, \dots, m, k = 1, \dots, n \tag{3}$$

$$\sum_{\{j=1,\dots,m \mid z-h_j\ge 0\}}\ \sum_{k'=1}^{k}\ \sum_{\{x''\in X_j \mid x-l_j+1\le x''\le x+l_i-1\}}\ \sum_{\{y''\in Y_j \mid y-w_j+1\le y''\le y+w_i-1\}} \mathrm{L}_{ij}^{[1]} \cdot \mathrm{W}_{ij}^{[1]} \cdot a_{jk'x''y''(z-h_j)}$$
$$\ge l_i \cdot w_i \cdot a_{ikxyz} \tag{4}$$
$$\text{where} \begin{cases} \mathrm{L}_{ij}^{[1]} = \min(x + l_i, x'' + l_j) - \max(x, x''), & i = 1, \dots, m,\ k = 1, \dots, n \\ \mathrm{W}_{ij}^{[1]} = \min(y + w_i, y'' + w_j) - \max(y, y''), & x \in X_i,\ y \in Y_i,\ z \in Z_i \backslash \{0\} \end{cases}$$

$$(x + l_i) \cdot a_{ikxyz} \le L'_k, \quad i = 1, \dots, m,\ k = 1, \dots, n$$
$$x \in X_i,\ y \in Y_i,\ z \in Z_i \tag{5}$$

$$L'_{k-1} - \delta_{ik} \le x \cdot a_{ikxyz} + (1 - a_{ikxyz}) \cdot M, \quad i = 1, \ldots, m,\ k = 2, \ldots, n \quad x \in X_i, y \in Y_i,\ z \in Z_i \tag{6}$$

$$L'_{k-1} \le L'_k, \quad k = 2, \ldots, n \tag{7}$$

$$L'_k \ge 0, \quad k = 1, \ldots, n \tag{8}$$

$$a_{ikxyz} \in \{0, 1\}, \quad i = 1, \ldots, m,\ k = 1, \ldots, n \quad x \in X_i,\ y \in Y_i,\ z \in Z_i \tag{9}$$

where X, Y, Z and X_i, Y_i, Z_i are defined as before. In this formulation, the objective function (1) aims to minimize the necessary length L'_n in order to load all boxes of all destinations, constraints (2) ensure that there are no overlap among the boxes inside the container, and constraints (3) ensure that the number of boxes (b_{ik}) of type i required by destination (customer) k is loaded in the container (note that $\sum_{k=1}^{n} b_{ik} = b_i$). Constraints (4) refer to the vertical stability of the cargo and ensure that the area of the bottom face of a box of type i must be completely supported (i.e., 100% supported) by the area of the top faces of one or more boxes placed immediately below them, or by the container's floor. It is worth noting that in these constraints, the boxes of a certain destination must be packed either over the boxes of the same destination or over the boxes of a destination that will be visited later in the sequence, so that the unloading of the boxes is not hampered. Constraints (5) ensure that boxes of destination k are packed within the length limit L'_k, constraints (6) ensure that boxes of destination k cannot be loaded δ_{ik} units of length beyond the necessary length L'_{k-1}, and constraints (7) ensure that the necessary length to pack all boxes of destination $k-1$ is smaller than the minimum length necessary to pack all boxes of destination k. Finally, constraints (8) and (9) define the domain of the decision variables. We note that model (1)–(9) can be seen as a strip-packing formulation for the container loading problem with multi-drop constraints.

In case there is not enough space to pack all boxes of all destinations inside the container (i.e., the container length L is not sufficiently large and some boxes may be left out of the loading), model (1)–(9) can be simply modified to maximize the total volume (or value) of the boxes packed. To this end, the objective function (1) should be replaced by:

$$\max \quad \sum_{i=1}^{m} \sum_{k=1}^{n} \sum_{x \in X_i} \sum_{y \in Y_i} \sum_{z \in Z_i} v_i \cdot a_{ikxyz} \tag{1a}$$

where v_i is the value of a box of type i (if $v_i = (l_i \cdot w_i \cdot h_i)$, then (1a) maximizes the total volume of the boxes), constraints (3) should be rewritten as the inequality:

$$\sum_{x \in X_i} \sum_{y \in Y_i} \sum_{z \in Z_i} a_{ikxyz} \le b_{ik}, \quad i = 1, \ldots, m,\ k = 1, \ldots, n \tag{3a}$$

and constraints (8) should be changed to:

$$0 \le L'_k \le L, \quad k = 1, \ldots, n \tag{8a}$$

We note in this model that L'_k becomes the length to load not necessarily all boxes of destinations $1, 2, \ldots, k$, but the most valuable boxes of these destinations, as boxes of one or more destinations may be left out of the loading. Therefore, this formulation based on (1a), (3a) and (8a) models the single container loading problem with multi-drop constraints.

As pointed out in Christofides and Whitlock (1977) and Beasley (1985), for a given cutting or packing pattern, each packed box could be moved down and/or forward and/or to the left, until its bottom, front and left-hand face are adjacent to other boxes or to the container. Without loss of generality in different cutting and packing problems, these patterns, called normal patterns, enabled us to restrict the sets X, Y and Z to:

$$X = \left\{ x | x = \sum_{i=1}^{m} \varepsilon_i \cdot l_i,\ 0 \le x \le L - \min_i(l_i), 0 \le \varepsilon_i \le b_i \text{ and integer},\ i = 1, \ldots, m \right\} \tag{10}$$

$$Y = \left\{ y | y = \sum_{i=1}^{m} \varepsilon_i \cdot w_i,\ 0 \le y \le W - \min_i(w_i),\ 0 \le \varepsilon_i \le b_i \text{ and integer},\ i = 1, \ldots, m \right\} \tag{11}$$

$$Z = \left\{ z | z = \sum_{i=1}^{m} \varepsilon_i \cdot h_i,\ 0 \le z \le H - \min_i(h_i),\ 0 \le \varepsilon_i \le b_i \text{ and integer},\ i = 1, \ldots, m \right\} \tag{12}$$

However, this is not the case for the present packing problem. It can be shown that there is loss of generality if $0 < \delta_{ik} < L$ and if the original sets X, Y and Z of model (1)–(9) are reduced as in (10)–(12) (normal patterns). Firstly, we must remember that the use of the normal patterns "tries to pull" the boxes as much as possible towards the left (the rear) in the container. On the other hand, the use of a positive value for parameter δ_{ik} "prevents" the boxes from being pulled as much as possible towards the left in the container, and we cannot ensure that there will be a coordinate along axis x where a box can be placed, which violates the basic assumption of using normal patterns.

One way to overcome this problem is to define parameter δ_{ik} as a multiple of the length l_i, for all i, i.e., $\delta_{ik} = \eta \cdot l_i$, with η integer. Note that this assumption will produce conservative solutions, as it is always sure that the boxes will be promptly available when each drop-off point is reached. Additionally, we need to rewrite the possible positions along axis x subtracting the normal patterns from the last coordinate originally generated, i.e., to consider that one box could be moved to the right in the container, until its right-hand face is adjacent to the last coordinate along axis x. The possible positions along axes y and z need not be changed. Therefore, the set X in (10) should be redefined as:

$$X = \left\{ x | 0 \le x = x' - \sum_{i=1}^{m} \varepsilon_i \cdot l_i,\ 0 \le \varepsilon_i \le b_i \text{ and integer},\ i = 1, \ldots, m \right\} \tag{13}$$

where $x' = \max\{x | x = \sum_{i=1}^{m} \varepsilon_i \cdot l_i \le L, 0 \le \varepsilon_i \le b_i \text{ and integer},\ i = 1, \ldots, m\}$ and the subsets X_i, Y_i, Z_i are now defined as a function of sets X, Y and Z in (11)–(13). Note that rewriting (10) as (13) is always necessary when $0 < \delta_{ik} < L$, but it is not necessary when $\delta_{ik} = 0$. Figure 4 shows an illustrative example of this situation for a container with size $(L, W, H) = (12, 1, 2)$ and two types of boxes $(l_1, w_1, h_1) = (9, 1, 1)$ (from destination 1) with $b_{11} = 1$, and $(l_2, w_2, h_2) = (4, 1, 1)$ (from destination 2) with $b_{22} = 1$. Here we are interested in showing that there is a lack of normal patterns along the x axis for a given $\delta_{ik} = 4$, for all i and all k. Note that if set X were used as in (10) (i.e., $X = \{0, 4\}$) (Fig. 4, left), there would not be any coordinates along the x axis where the box of destination 2 could be

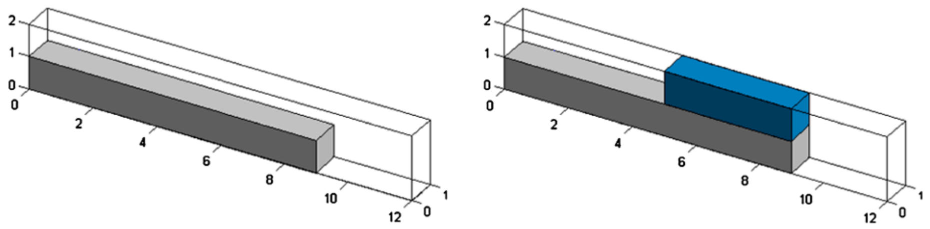


Fig. 4 Solutions produced using sets X, Y and Z as in (10)–(12) and as in (11)–(13), respectively

placed. It means that this box will be placed after box of destination 1, i.e., it should be placed out of the container, since it does not fit inside the container. Note, however, that using set X as in (13) (i.e., $X = \{0, 5\}$) (Fig. 4, right), we can overcome this problem.

Note that model (1)–(9), or its modified version using (1a), (3a) and (8a), consists of $\sum_{i=1}^{m}\sum_{k=1}^{n}|X_i|\cdot|Y_i|\cdot|Z_i|$ binary variables plus n continuous variables, besides $|X|\cdot|Y|\cdot|Z| + m\cdot n + \sum_{i=1}^{m}\sum_{k=1}^{n}|X_i|\cdot|Y_i|\cdot(|Z_i|-1) + \sum_{i=1}^{m}\sum_{k=1}^{n}|X_i|\cdot|Y_i|\cdot|Z_i| + \sum_{i=1}^{m}\sum_{k=2}^{n}|X_i|\cdot|Y_i|\cdot|Z_i| + n - 1$ constraints. It is worth mentioning that these models could be easily adapted to also consider the horizontal stability of the cargo (by means of additional constraints very related to (4); Junqueira et al. (2012) and other constraints, such as the weight limit of the cargo.

3 Solution approaches considering multi-dropping

Based on the models presented in Sect. 2 (with X, Y and Z defined as in (11)–(13)), we develop two approaches to address the container loading problem considering multi-dropping. The first approach assumes that parameter δ_{ik}, relative to the reach of the worker responsible for arranging the boxes of destination k into a container, is equal to 0, whereas the second approach assumes that $0 < \delta_{ik} < L$. In both approaches, first we minimize the length L' of a hypothetical container (L', W, H) necessary to pack all the boxes from all destinations. If the minimum necessary length L' is larger than L, then we apply a simple greedy heuristic to maximize the total volume (or value) of the boxes packed inside the container (L, W, H). It is worth mentioning that there is no motivation for developing an approach for the case where $\delta_{ik} = L$, as the resulting loading pattern would not take into account the different destinations of the boxes.

3.1 Case in which $\delta_{ik} = 0$: approach 1

This situation represents a more conservative case, where the worker is not allowed to go inward of the "border" between the boxes of different destinations, i.e., the worker cannot take advantage of any empty spaces fortuitously produced by the boxes of the earlier destinations to pack the boxes of a coming destination (see Fig. 3). This allows us to decompose, without loss of generality, the strip-packing model (1)–(9) of minimizing the necessary container length L' of the hypothetical container (L', W, H) to pack all boxes of all n destinations, into n independent submodels. Each one of these submodels consists of minimizing the necessary length to pack all boxes of each destination k, and the final solution of the problem is a composition of the n solutions of the corresponding submodels. Note that although this approach does not take advantage of the empty spaces fortuitously produced, it makes cargo handling easier when each drop-off point is reached. Indeed, there are Brazilian carriers that use special devices (mobile subdivisions like “curtains”) in their containers or trucks to physically separate the cargo loaded in each container by its destination.

Since the problem can be decomposed, it is possible to redefine the possible coordinates where the front-left-bottom corner of a box can be placed. For each destination k, the possible positions along axes x, y and z of the container, respectively, belong to the sets $X_k = \{0, 1, 2, \ldots, L' - \min_i(l_i)\}$, $Y_k = \{0, 1, 2, \ldots, W - \min_i(w_i)\}$ and $Z_k = \{0, 1, 2, \ldots, H - \min_i(h_i)\}$, $i \in M_k$, where $M_k = \{i = 1, \ldots, m | b_{ik} > 0\}$, i.e., the subset of box types i that must be unloaded in destination k. Since $\delta_{ik} = 0$, without loss of generality, we can restrict the sets X, Y and Z to the normal patterns:

$$X_k = \left\{ x | x = \sum_{i \in M_k} \varepsilon_i \cdot l_i,\ 0 \le x \le L' - \min_i(l_i),\ 0 \le \varepsilon_i \le b_i \text{ and integer},\ i \in M_k \right\},$$
$$k = 1, \ldots, n \tag{14}$$

$$Y_k = \left\{ y | y = \sum_{i \in M_k} \varepsilon_i \cdot w_i,\ 0 \le y \le W - \min_i(w_i),\ 0 \le \varepsilon_i \le b_i \text{ and integer},\ i \in M_k \right\},$$
$$k = 1, \ldots, n \tag{15}$$

$$Z_k = \left\{ z | z = \sum_{i \in M_k} \varepsilon_i \cdot h_i,\ 0 \le z \le H - \min_i(h_i),\ 0 \le \varepsilon_i \le b_i \text{ and integer},\ i \in M_k \right\}$$
$$k = 1, \ldots, n \tag{16}$$

i.e., we do not need to rewrite the possible positions along axis x, subtracting the original coordinates from the last coordinate originally generated, as we did at the end of Sect. 2. Defining $X_{ik} = \{x \in X_k | 0 \le x \le L' - l_i\}$, $Y_{ik} = \{y \in Y_k | 0 \le y \le W - w_i\}$ and $Z_{ik} = \{z \in Z_k | 0 \le z \le H - h_i\}$, $i \in M_k$ and $k = 1, \ldots, n$, as subsets of X_k, Y_k and Z_k in (14)–(16), respectively, and based on model (1)–(9) and its modified version with (1a), (3a) and (8a), we propose the following two-part iterative procedure in k (Approach 1):

Part A
For $k = 1, \ldots, n$, solve the strip-packing formulation (17)–(23):

$$\min \quad L'_k \tag{17}$$

$$\sum_{i=1}^{m} \sum_{\{x \in X_{ik} | x' - l_i + 1 \le x \le x'\}} \sum_{\{y \in Y_{ik} | y' - w_i + 1 \le y \le y'\}} \sum_{\{z \in Z_{ik} | z' - h_i + 1 \le z \le z'\}} a_{ikxyz} \le 1,$$
$$x' \in X_k,\ y' \in Y_k,\ z' \in Z_k \tag{18}$$

$$\sum_{x \in X_{ik}} \sum_{y \in Y_{ik}} \sum_{z \in Z_{ik}} a_{ikxyz} = b_{ik} \quad i = 1, \ldots, m \tag{19}$$

$$\sum_{\{j=1,\ldots,m | z - h_j \ge 0\}} \sum_{\{x'' \in X_{jk} | x - l_j + 1 \le x'' \le x + l_i - 1\}} \sum_{\{y'' \in Y_{jk} | y - w_j + 1 \le y'' \le y + w_i - 1\}} \mathrm{L}_{ij}^{[1]} \cdot \mathrm{W}_{ij}^{[1]} \cdot a_{jkx''y''(z-h_j)}$$
$$\ge l_i \cdot w_i \cdot a_{ikxyz} \tag{20}$$

$$\text{where} \begin{cases} \mathrm{L}_{ij}^{[1]} = \min(x + l_i, x'' + l_j) - \max(x, x''), & i = 1, \ldots, m \\ \mathrm{W}_{ij}^{[1]} = \min(y + w_i, y'' + w_j) - \max(y, y''), & x \in X_{ik},\ y \in Y_{ik},\ z \in Z_{ik} \backslash \{0\} \end{cases}$$

$$(x + l_i) \cdot a_{ikxyz} \le L'_k, \quad i = 1, \ldots, m$$
$$x \in X_{ik},\ y \in Y_{ik},\ z \in Z_{ik} \tag{21}$$
$$L'_k \ge 0, \quad i = 1, \ldots, m \tag{22}$$
$$a_{ikxyz} \in \{0, 1\}, \quad i = 1, \ldots, m$$
$$x \in X_{ik},\ y \in Y_{ik},\ z \in Z_{ik} \tag{23}$$

Return $L'^*_1, L'^*_2, \ldots, L'^*_n$ (i.e., the minimum lengths from iterations $1, 2, \ldots, n$).

If $L' = \sum_{k=1}^{n} L'^*_k \le L$, then the solution packs all the required boxes in (L, W, H). Otherwise,

Part B

For $k = 1, \ldots, n$, subtract the difference $L' - L$ from L'^*_k (provided that $L'^*_k > L' - L$) and solve n additional single container loading models (24)–(28) with container sizes $(L'^*_k - (L' - L), W, H)$:

$$\max \quad \sum_{i=1}^{m} \sum_{x \in X_i} \sum_{y \in Y_i} \sum_{z \in Z_i} v_i \cdot a_{ikxyz} \tag{24}$$

$$\sum_{i=1}^{m} \sum_{\{x \in X_{ik} \mid x' - l_i + 1 \le x \le x'\}} \sum_{\{y \in Y_{ik} \mid y' - w_i + 1 \le y \le y'\}} \sum_{\{z \in Z_{ik} \mid z' - h_i + 1 \le z \le z'\}} a_{ikxyz} \le 1,$$
$$x' \in X_k,\ y' \in Y_k,\ z' \in Z_k \tag{25}$$

$$\sum_{x \in X_{ik}} \sum_{y \in Y_{ik}} \sum_{z \in Z_{ik}} a_{ikxyz} \le b_{ik}, \quad i = 1, \ldots, m \tag{26}$$

$$\sum_{\{j=1,\ldots,m \mid z - h_j \ge 0\}} \sum_{\{x'' \in X_{jk} \mid x - l_j + 1 \le x'' \le x + l_i - 1\}} \sum_{\{y'' \in Y_{jk} \mid y - w_j + 1 \le y'' \le y + w_i - 1\}} \mathrm{L}_{ij}^{[1]} \cdot \mathrm{W}_{ij}^{[1]} \cdot a_{jkx''y''(z-h_j)}$$
$$\ge l_i \cdot w_i \cdot a_{ikxyz} \tag{27}$$
$$\text{where} \begin{cases} \mathrm{L}_{ij}^{[1]} = \min(x + l_i, x'' + l_j) - \max(x, x''), & i = 1, \ldots, m \\ \mathrm{W}_{ij}^{[1]} = \min(y + w_i, y'' + w_j) - \max(y, y''), & x \in X_{ik},\ y \in Y_{ik},\ z \in Z_{ik} \backslash \{0\} \end{cases}$$

$$a_{ikxyz} \in \{0, 1\}, \quad i = 1, \ldots, m,$$
$$x \in X_{ik}, y \in Y_{ik},\ z \in Z_{ik} \tag{28}$$

where sets X_k and X_{ik} are properly redefined to consider the possible positions along axis x using $L'^*_k - (L' - L)$ and not L' anymore.

Then choose the "combined solution" (composed of the kth solution of Part B and the remaining $n - 1$ solutions of Part A) that results in the maximum total value (or volume) of the boxes packed inside container (L, W, H).

If $L'^*_k \le L' - L$ for all k, then solve the 0–1 knapsack model (29)–(31):

$$\max \quad \sum_{k=1}^{n} v_k \cdot e_k \tag{29}$$

$$\sum_{k=1}^{n} L_k'^* \cdot e_k \le L \tag{30}$$

$$e_k \in \{0,1\}, \quad k = 1,\dots,n \tag{31}$$

i.e., exclude all boxes of the destination(s) that contribute(s) with the lowest volume (or value).

This is the (greedy heuristic) solution returned by this procedure to the original problem.

Note in Part A that index k is fixed in each iteration and, therefore, the objective function (17) aims to minimize the necessary length L_k' to pack all boxes from destination k only (and not from all destinations $1,2,\dots,n$, as in model (1)–(9)). Therefore, constraints (6) and (7) of model (1)–(9) do not appear in model (17)–(23). In other words, each model (17)–(23) in k is independent from the others in $1,2,\dots,k-1,k+1,\dots,n$. Thus, it is expected that model (17)–(23) in Part A becomes easier to solve than model (1)–(9), which has greater dimensions in terms of the number of variables and constraints. In Part B, we solve single container loading models to maximize the total volume (or value) of the boxes packed of each destination at once (i.e., boxes of only one destination may be left out of the loading). Note that this combined solution is composed of the solutions of $n-1$ models (17)–(23) solved in Part A, plus the solution of either one of the n additional models (24)–(28) (when $L_k'^* > L' - L$) or one additional model (29)–(31) (when $L_k'^* \le L' - L$ for all k) solved in Part B.

Figure 5 shows a possible loading pattern, using this procedure, with boxes from three different destinations. Note that a kind of (imaginary) boundary curtain is created between the boxes of two consecutive destinations inside the container. As mentioned before, the existence of mobile subdivisions (producing compartments of variable sizes) when loading a container or truck is common practice in some Brazilian delivery companies.

Note that the strip-packing model (17)–(23) in Part A consists, for each destination k, of $|M_k|\cdot|X_{ik}|\cdot|Y_{ik}|\cdot|Z_{ik}|$ binary variables plus 1 continuous variable, besides $|X_k|\cdot|Y_k|\cdot|Z_k| + |M_k| + |M_k|\cdot|X_{ik}|\cdot|Y_{ik}|\cdot(|Z_{ik}|-1) + |M_k|\cdot|X_{ik}|\cdot|Y_{ik}|\cdot|Z_{ik}|$ constraints. In Part B, the single container loading model (24)–(28) consists, for each destination k, of $|M_k|\cdot|X_{ik}|\cdot|Y_{ik}|\cdot|Z_{ik}|$ binary variables plus $|X_k|\cdot|Y_k|\cdot|Z_k| + |M_k| + |M_k|\cdot|X_{ik}|\cdot|Y_{ik}|\cdot(|Z_{ik}|-1)$ constraints, and the 0–1 knapsack model (29)–(31) consists of n binary variables plus 1 constraint.

3.2 Case in which $0 < \delta_{ik} < L$: approach 2

This is the more general case, where the worker is allowed to go only partially inward of the "border" between the boxes of different destinations, i.e., the worker can take advantage of some empty spaces fortuitously produced by the boxes of earlier destinations to pack the boxes of a coming destination (see Fig. 3). Note that in this approach, we aim to take advantage of the empty spaces fortuitously produced without compromising cargo handling when each drop-off point is reached.

In this case, as $0 < \delta_{ik} < L$, we shall use sets X, Y and Z as defined in (11)–(13), i.e., we need to rewrite the possible positions along axis x subtracting the original coordinates from the last coordinate originally generated. Based on model (1)–(9) and its modified version with (1a), (3a) and (8a), we propose the following two-part iterative procedure in k (Approach 2):

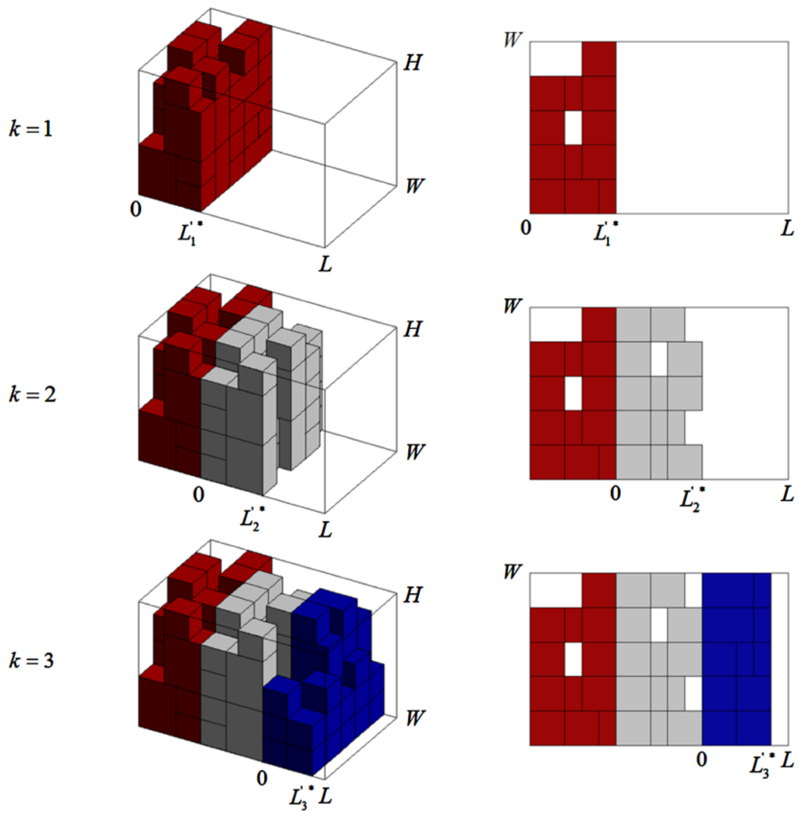


Fig. 5 Example of a loading pattern obtained by approach 1 with $\delta_{ik} = 0$

Part A

Set $k = 1$, and solve the strip-packing formulation (32)–(39):

$$\min \quad L'_k \tag{32}$$

$$\sum_{i=1}^{m}\sum_{k'=1}^{k}\sum_{\{x\in X_i|x'-l_i+1\le x\le x'\}}\sum_{\{y\in Y_i|y'-w_i+1\le y\le y'\}}\sum_{\{z\in Z_i|z'-h_i+1\le z\le z'\}} a_{ik'xyz}\le 1,$$
$$x'\in X,\ y'\in Y,\ z'\in Z \tag{33}$$

$$\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} a_{ikxyz} = b_{ik}, \quad i=1,\dots,m \tag{34}$$

$$\sum_{\{j=1,\dots,m|z-h_j\ge 0\}}\sum_{k'=1}^{k}\sum_{\{x''\in X_j|x-l_j+1\le x''\le x+l_i-1\}}\sum_{\{y''\in Y_j|y-w_j+1\le y''\le y+w_i-1\}} \mathrm{L}_{ij}^{[1]}\cdot \mathrm{W}_{ij}^{[1]}\cdot a_{jk'x''y''(z-h_j)}$$
$$\ge l_i\cdot w_i\cdot a_{ikxyz} \tag{35}$$

$$\text{where}\begin{cases}\mathrm{L}_{ij}^{[1]} = \min(x+l_i, x''+l_j)-\max(x,x''), & i=1,\dots,m,\\ \mathrm{W}_{ij}^{[1]} = \min(y+w_i, y''+w_j)-\max(y,y''), & x\in X_i,\ y\in Y_i,\ z\in Z_i\backslash\{0\}\end{cases}$$

$$(x + l_i) \cdot a_{ikxyz} \le L'_k, \quad i = 1, \dots, m, \quad x \in X_i,\ y \in Y_i,\ z \in Z_i \tag{36}$$

$$L'_{k-1} - \delta_{ik} \le x \cdot a_{ikxyz} + (1 - a_{ikxyz}) \cdot M, \quad i = 1, \dots, m, \quad x \in X_i,\ y \in Y_i, z \in Z_i \tag{37}$$

$$L'_k \ge 0 \tag{38}$$

$$a_{ikxyz} \in \{0, 1\}, \quad i = 1, \dots, m, \quad x \in X_i, y \in Y_i,\ z \in Z_i \tag{39}$$

Fix variables $a^*_{ikxyz} = 1$ and L'^*_k relative to the solution of model (32)–(39) for destination k.

Set $k = k + 1$ and solve the model above in k, with the variables a_{ikxyz} and L'_k above fixed, for $k' = 1, 2, \dots, k - 1$.

Repeat this procedure for all n destinations.

Return the last L'^*_n obtained.

If $L' = L'^*_n \le L$, then the solution packs all the required boxes in (L, W, H). Otherwise,

Part B

Maintain fixed the variables $a^*_{ikxyz} = 1$ relative to the boxes of destinations $k' = 1, 2, \dots, k$ in which $L'^*_k < L$, and solve for the boxes of the remaining destinations $k'' = k+1, k+2, \dots, n$ the additional single container loading model (40)–(49) with container size (L, W, H):

$$\max \sum_{i=1}^{m} \sum_{k''=k+1}^{n} \sum_{x\in X_i} \sum_{y\in Y_i} \sum_{z\in Z_i} v_i \cdot a_{ik''xyz} \tag{40}$$

$$\sum_{i=1}^{m} \sum_{k=1}^{n} \sum_{\{x\in X_i \mid x'-l_i+1\le x\le x'\}} \sum_{\{y\in Y_i \mid y'-w_i+1\le y\le y'\}} \sum_{\{z\in Z_i \mid z'-h_i+1\le z\le z'\}} a_{ikxyz} \le 1, \quad x' \in X,\ y' \in Y,\ z' \in Z \tag{41}$$

$$\sum_{x\in X_i} \sum_{y\in Y_i} \sum_{z\in Z_i} a_{ik'xyz} = b_{ik'}, \quad i = 1, \dots, m,\ k' = 1, \dots, k \tag{42}$$

$$\sum_{x\in X_i} \sum_{y\in Y_i} \sum_{z\in Z_i} a_{ik''xyz} \le b_{ik''}, \quad i = 1, \dots, m,\ k'' = k + 1, \dots, n \tag{43}$$

$$\sum_{\{j=1,\dots,m \mid z-h_j\ge 0\}} \sum_{k'=1}^{k''} \sum_{\{x''\in X_j \mid x-l_j+1\le x''\le x+l_i-1\}} \sum_{\{y''\in Y_j \mid y-w_j+1\le y''\le y+w_i-1\}} \mathrm{L}^{[1]}_{ij} \cdot \mathrm{W}^{[1]}_{ij} \cdot a_{jk'x''y''(z-h_j)} \ge l_i \cdot w_i \cdot a_{ik''xyz} \tag{44}$$

$$\text{where} \begin{cases} \mathrm{L}^{[1]}_{ij} = \min(x + l_i, x'' + l_j) - \max(x, x''), & i = 1, \dots, m,\ k'' = k + 1, \dots, n \\ \mathrm{W}^{[1]}_{ij} = \min(y + w_i, y'' + w_j) - \max(y, y''), & x \in X_i,\ y \in Y_i,\ z \in Z_i \backslash \{0\} \end{cases}$$

$$(x + l_i) \cdot a_{ik''xyz} \le L'_{k''}, \quad i = 1, \dots, m,\ k'' = k + 1, \dots, n, \quad x \in X_i,\ y \in Y_i,\ z \in Z_i \tag{45}$$

$$L'_{k''-1} - \delta_{ik''} \leq x \cdot a_{ik''xyz} + (1 - a_{ik''xyz}) \cdot M, \quad i = 1, \dots, m,\ k'' = k+1, \dots, n$$
$$x \in X_i,\ y \in Y_i,\ z \in Z_i \tag{46}$$
$$L'_{k''-1} \leq L'_{k''}, \quad k'' = k+1, \dots, n \tag{47}$$
$$0 \leq L'_{k''} \leq L, \quad k'' = k+1, \dots, n \tag{48}$$
$$a_{ikxyz} \in \{0, 1\}, \quad i = 1, \dots, m,\ k = 1, \dots, n$$
$$x \in X_i, y \in Y_i, z \in Z_i \tag{49}$$

This is the (greedy heuristic) solution returned by this procedure to the original problem.

Note in Part A that index k is fixed in each iteration, and, therefore, the objective function (32) aims to minimize the necessary length L'_k to pack all boxes of destinations $1, 2, \dots, k$, in which the solutions from model (32)–(39) for the earlier iterations $k' = 1, 2, \dots, k-1$ are fixed in $a^*_{ikxyz} = 1$ and L'^*_k in iteration k. Note also that constraints (7) of model (1)–(9) do not appear in model (32)–(39), since the procedure is iterative (therefore, $L'_1 \leq L'_2 \leq \dots \leq L'_n$). It is worth noting that this procedure is different from the procedure where $\delta_{ik} = 0$ (Approach 1), since each model (32)–(39) fixed in k depends on the earlier models in $k-1, k-2, \dots, 1$. This procedure performs a myopic optimization for the boxes of each destination k, as the boxes of each destination, once fixed, cannot be rearranged later, which could lead to a loss of the optimal solution, instead of considering the optimization of all boxes of all n destinations at once.

In Part B, we solve a single container loading model (40)–(49) to maximize the total volume (or value) of the boxes packed of the destinations in which $L'^*_k > L$ (i.e., boxes of one or more of these destinations may be left out of the loading). We observe that constraints (42) refer to the boxes (with the related variables fixed) of the destinations $k' = 1, \dots, k$ in which $L'^*_{k'} < L$ (solved in Part A), and constraints (43) refer to the boxes of destinations $k'' = k+1, \dots, n$ in which $L'^*_{k''} > L$. The combined solution is composed of the solution of one model solved in Part A plus the solution of one additional model solved in Part B.

Figure 6 shows a possible loading pattern, using this procedure with $\delta_{ik} = l_i$, with boxes from three different destinations. Note that the loading pattern is like a three-dimensional version of the game Tetris (2009), developed in the eighties, where the objective is to fit two-dimensional pieces (polyominoes) of different shapes that fall from the top of a computer screen. When a line on the screen is complete, this line disappears and extra points are given to the player. The game ends when the incomplete lines are heaped up until the top of the screen. In fact, a disadvantage of the procedure described above, with respect to model (1)–(9), is that the former ignores which are the coming boxes that must be loaded inside the container, and, therefore, it adopts a greedy posture with respect to the boxes of the present destination.

Note that the strip-packing model (32)–(39) in Part A consists, for each destination k, of $\sum_{i=1}^{m} |X_i| \cdot |Y_i| \cdot |Z_i|$ binary variables plus 1 continuous variable, besides $|X| \cdot |Y| \cdot |Z| + m + \sum_{i=1}^{m} |X_i| \cdot |Y_i| \cdot (|Z_i| - 1) + 2\sum_{i=1}^{m} |X_i| \cdot |Y_i| \cdot |Z_i|$ constraints. In Part B, the single container loading model (40)–(49) consists of $\sum_{i=1}^{m} \sum_{k=1}^{n} |X_i| \cdot |Y_i| \cdot |Z_i|$ binary variables and $\sum_{k''=k+1}^{n} 1$ continuous variables, plus $|X| \cdot |Y| \cdot |Z| + \sum_{k'=1}^{k} m + \sum_{k''=k+1}^{n} m + \sum_{i=1}^{m} \sum_{k''=k+1}^{n} |X_i| \cdot |Y_i| \cdot (|Z_i| - 1) + 2\sum_{i=1}^{m} \sum_{k''=k+1}^{n} |X_i| \cdot |Y_i| \cdot |Z_i| + \sum_{k''=k+1}^{n} 1$ constraints.

As previously noted, Part A of this procedure performs a myopic optimization for the boxes of each destination k. However, we can "aid" this procedure to produce more accessible empty spaces where boxes of coming destinations may be placed, respecting the limit

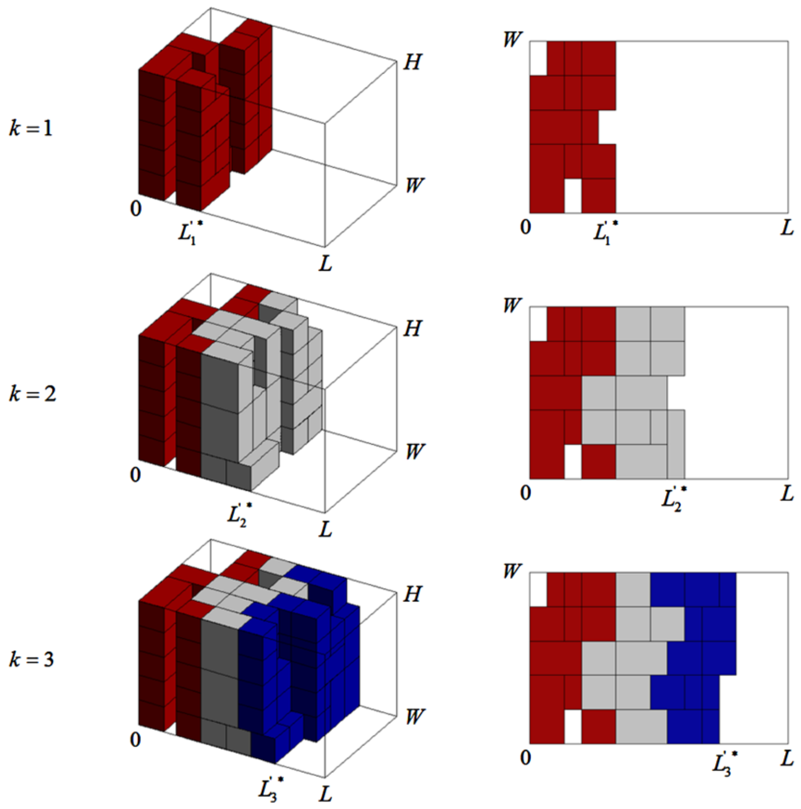


Fig. 6 Example of a loading pattern obtained by approach 2 with $\delta_{ik} = l_i$

imposed by parameter δ_{ik}, relative to the reach of the worker. The idea consists of adding a "breaking tie" term to the objective function (32), with a fractionary value between 0 and 1, in such a way that the resulting loading pattern favors loadings where the boxes of a destination are placed as far back as possible inside the container. In such a way, the boxes of the coming destinations can be easily arranged, without hindering cargo handling. Note, however, that the modified objective function (50) is not integral anymore.

$$\min \quad L_k' + \frac{\sum_{i=1}^{m}\sum_{k'=1}^{k}\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} x\cdot a_{ik'xyz}}{(\sum_{i=1}^{m}\sum_{k'=1}^{k}\sum_{x\in X_i}\sum_{y\in Y_i}\sum_{z\in Z_i} x)+1} \tag{50}$$

Figure 7 shows a possible loading pattern, using this procedure with $\delta_{ik} = l_i$ and the breaking tie function, with boxes of three different destinations.

4 Computational results

Model (1)–(9) and its modified version with (1a), (3a) and (8a) of Sect. 2 and approaches 1 and 2 of Sect. 3 were implemented in the modeling language GAMS (version 22.7) and the solver CPLEX 11.0 (with default parameters) was used to solve them. All computational tests were performed in a PC Core i7 (2.8 GHz, 8.0 GB). To evaluate their performances,

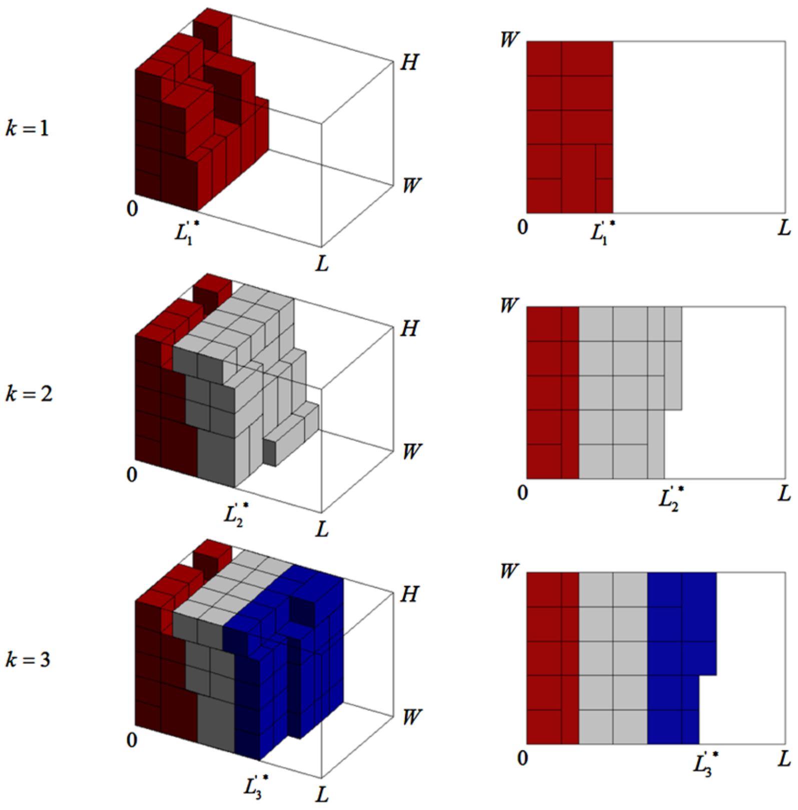


Fig. 7 Example of a loading pattern obtained by approach 2 with $\delta_{ik} = l_i$ and the breaking tie function

the models and the approaches were tested with randomly generated instances and instances from the literature.

4.1 Results with randomly generated instances

The following parameters were used in the randomly generated instances:

- Four types of boxes: $m = 1$ (in this case, the boxes can rotate around all axes), $m = 5$, $m = 10$ and $m = 20$ (in these three cases, the boxes have fixed orientation).
- Boxes dimensions generated in two different ways: (A_m, $m = 1, 5, 10$ and 20) with box dimensions varying between 25% and 75% of the dimensions of the container, i.e., $l_i \in [0.25L, 0.75L]$, $w_i \in [0.25W, 0.75W]$ and $h_i \in [0.25H, 0.75H]$; (B_m, $m = 1, 5, 10$ and 20) with box dimensions varying between 10% and 50% of the container dimensions, i.e, $l_i \in [0.10L, 0.50L]$, $w_i \in [0.10W, 0.50W]$ and $h_i \in [0.10H, 0.50H]$. The values v_i of the boxes were considered as their respective volumes $l_i \cdot w_i \cdot h_i$. For the sake of simplicity, in all examples we consider cubic containers, i.e., with dimensions $L = W = H$.

When $m = 1$, an additional decision variable for each possible box orientation was defined, resulting in a total of six decision variables, and the MIP-based approaches were properly modified to consider these new variables. An alternative to handle this case would be to consider each of the six possible rotations of a box as a different box type, i.e., $m = 6$, and to limit the maximum number of boxes loaded in constraints (3), (19), (26), (34), (42) and (43).

Table 1 Number of boxes, number of elements of the normal pattern sets and number of constraints and variables of model (1)–(9) and approaches 1 and 2 (Part A only) in examples A_m and B_m

	No boxes	No normal patterns $\|X\|$	$\|Y\|$	$\|Z\|$	No var. Model (1)–(9)	Approach where $\delta_{ik}=0$	$\delta_{ik}=l_i$	No con. Model (1)–(9)	Approach where $\delta_{ik}=0$	$\delta_{ik}=l_i$
A_1	20	8	6	6	1648	550	550	4139	1131	1679
A_5	41	14	10	10	7493	2563	3093	19804	5806	9879
A_{10}	99	8	6	10	2349	1494	1494	6138	3043	4535
A_{20}	89	15	6	6	2674	3814	4147	6909	7083	11814
B_1	500	16	11	11	24604	8202	8202	64556	17052	25252
B_5	813	16	11	11	10048	4937	4937	27150	10807	15742
B_{10}	1000	16	11	11	9004	10004	10004	24609	20379	30381
B_{20}	674	16	11	11	22384	20352	20352	58866	39845	60195

For the sake of simplicity, and only for the generation of the box dimensions and its available amount b_i, we have considered containers with $L = W = H = 10$. However, the container dimensions considered in the computational tests were either $L = 15$ or $L = 12$ and $W = H = 10$. The number n of destinations was set to 3. The amount b_i of available boxes of type i was defined as $b_i = \lfloor (L \cdot W \cdot H)/(l_i \cdot w_i \cdot h_i) \rfloor$ for instances with $m = 1$, and it was randomly generated by a uniform distribution in the interval $[1, \lfloor L/l_i \rfloor \cdot \lfloor W/w_i \rfloor \cdot \lfloor H/h_i \rfloor]$, for instances with $m = 5$, 10 and 20, $i = 1, \dots, m$. For the generation of the available amount b_{ik} of boxes of type i required by each destination k, the number of boxes reported in the optimal solution obtained with the base model presented in Junqueira et al. (2012) was randomly generated by a uniform distribution between the n destinations, such that $\sum_{k=1}^{n} b_{ik} = b_i$, $i = 1, \dots, m$. The value of the maximum reach of the worker δ_{ik} was arbitrarily set as 0 and l_i for each destination k. Note that, in cases where $\delta_{ik} = 0$ and $\delta_{ik} = l_i$, respectively, the approaches used were the ones described in Sects. 3.1 and 3.2.

Table 1 presents, for each instance, the total amount of boxes that must be loaded for all n destinations. In order to illustrate the size of the models generated using instances of groups A_m and B_m, this table also presents the size of sets X, Y and Z as in (11)–(13), and the number of constraints and binary variables, for each of the eight randomly generated instances. These numbers correspond to the values reported by CPLEX after pre-processing model (1)–(9) and approaches 1 and 2 (Part A only). It is worth noting that this number of variables and constraints presented in this table corresponds to the maximum value obtained among the n destinations. Note also that, as the cardinality of sets$X$, Y and Z increases, the number of variables and constraints also increases, and the solution of the models becomes significantly more difficult.

In the experiments that follow, the computational time spent to solve each model was limited to 1 hour (3600 seconds) and the optimality gaps were computed as:

$$\text{Gap} = \frac{\text{(best solution obtained—best bound obtained)}}{\text{(best bound obtained)}} 100\%$$

Therefore, four possible cases, with respect to the quality of the solution obtained by GAMS/CPLEX, can occur: (i) optimal solution, with Gap equal to zero; (ii) integer solution, with Gap greater than zero and with CPLEX exceeding the time limit; (iii) no solution, without Gap and with CPLEX exceeding the time limit; (iv) insufficient computer mem-

Table 2 Results obtained with model (1)–(9) (and its modified version with (1a), (3a) and (8a), when necessary) with $\delta_{ik} = 0$

	L	$L_1'^*$	$L_2'^*$	$L_3'^*$	$L_n'^*$	Gap (%)	Time (s)	Vol. (%)	No. boxes left out
A_1	15	4	8	14	14	0.000	3183.22	100.00	0
	12	2	6	12	12	5.000	3600.00	95.00	1
A_5	15	–	–	–	–	–	3600.00	–	–
A_{10}	15	4	8	12	12	0.000	2.27	100.00	0
A_{20}	15	–	–	–	–	–	3600.00	–	–
B_1	15	–	–	–	–	–	3600.00	–	–
B_5	15	7	11	15	15	49.573	3600.00	100.00	0
	12	4	8	12	12	0.000	1759.06	100.00	0
B_{10}	15	4	8	12	12	0.000	208.52	100.00	0
B_{20}	15	6	10	15	15	49.671	3600.00	100.00	0
	12	5	8	12	12	0.800	3600.00	99.20	2

ory to compile the model in GAMS, no Gap and no relevant information concerning the computational time. The last two cases are represented in the tables by the symbol "–".

The following tables show, for model (1)–(9) (and its modified version with (1a), (3a) and (8a), when necessary) and approaches 1 and 2 (Part A always, Part B when necessary), the length considered of the container and the minimum necessary length $L_k'^*$ to pack the boxes of each destination k. Note that, for approach 2 with $\delta_{ik} = l_i$, these values are the sum of the minimum necessary length $L_k'^*$ to pack all boxes of destination k, plus the boxes of the earlier destinations $1, 2, \dots, k-1$. Next, the tables show the values for the optimality Gap (in %), the runtime (in seconds) spent to solve each of the instances, the fraction of volume (in %) occupied by the boxes loaded in the container and the number of boxes left out of the container are also presented. For approach 1 with $\delta_{ik} = 0$, these values refer to the sum of the necessary values to solve the models for all n destinations. Remember that these models can be independently solved for each destination k (see Sect. 3.1).

The results obtained with model (1)–(9) (and its modified version with (1a), (3a) and (8a), when necessary) with $\delta_{ik} = 0$ and l_i for the randomly generated instances of groups A_m and B_m are presented in Tables 2 and 3. Note that, as expected, the total necessary length to pack all boxes of all destinations decreases as δ_{ik} increases from 0 to l_i. In particular, the model was able to solve to optimality instance B_1 with $\delta_{ik} = l_i$ (see Table 3), despite the great number of variables of this instance (see Table 1). Instances A_1, A_{10} and B_{10} were solved to optimality by the model with $\delta_{ik} = 0$ and l_i, while the model could not find any solution within the time limit in case of instance A_5. It is worth remembering that the related variables and constraints are all dependent among all n destinations. The modified version of the model with (1a), (3a) and (8a) was necessary for solving instances A_1, B_5 and B_{20} with $\delta_{ik} = 0$ (see Table 2), and instances B_5 and B_{20} with $\delta_{ik} = l_i$ (see Table 3). It is interesting to note that, for instance B_5 with $\delta_{ik} = 0$ and l_i, the modified version was able to find a solution that is optimal for both the modified version and for model (1)–(9), since no box was left out of the loading and all the boxes were packed considering the container length $L = 12$ (note that the solutions found by model (1)–(9) for this instance are suboptimal).

The results obtained for instances of groups A_m and B_m with the MIP-based approaches 1 and 2 with $\delta_{ik} = 0$ and l_i (Part A always, Part B when necessary), respectively, are presented in Tables 4, 5 and 6. Note that, unlike the results in Table 2, all solutions of approach 1

Table 3 Results obtained with model (1)–(9) (and its modified version with (1a), (3a) and (8a), when necessary) with $\delta_{ik} = l_i$

	L	$L_1'^*$	$L_2'^*$	$L_3'^*$	$L_n'^*$	Gap (%)	Time (s)	Vol. (%)	No. boxes left out
A_1	15	4	6	10	10	0.000	49.24	100.00	0
A_5	15	–	–	–	–	–	3600.00	–	–
A_{10}	15	4	8	10	10	0.000	4.46	100.00	0
A_{20}	15	5	8	10	10	0.000	1778.97	100.00	0
B_1	15	4	7	10	10	0.000	2555.49	100.00	0
B_5	15	7	11	15	15	50.000	3600.00	100.00	0
	12	4	8	12	12	0.000	1182.07	100.00	0
B_{10}	15	4	7	10	10	0.000	117.14	100.00	0
B_{20}	15	8	12	15	15	50.000	3600.00	100.00	0
	12	3	6	12	12	6.000	3600.00	94.00	21

Table 4 Results obtained with approach 1 with $\delta_{ik} = 0$

	L	$L_1'^*$	$L_2'^*$	$L_3'^*$	$L_n'^*$	Gap (%)	Time (s)	Vol. (%)	No. boxes left out
A_1	15	4	4	6	14	0.000	0.81	100.00	0
	12	2	4	6	12	0.000	0.283	95.00	1
A_5	15	5	5	5	15	0.000	349.14	100.00	0
	12	5	5	0	10	0.000	0.51	77.90	10
A_{10}	15	4	4	4	12	0.000	0.23	100.00	0
A_{20}	15	5	5	4	14	0.000	3.38	100.00	0
	12	5	5	2	12	0.000	0.271	92.80	3
B_1	15	4	4	4	12	0.000	35.74	100.00	0
B_5	15	4	4	4	12	0.000	11.88	100.00	0
B_{10}	15	4	4	4	12	0.000	1.23	100.00	0
B_{20}	15	4	4	4	12	0.000	66.82	100.00	0

were proved to be optimal. It is worth remembering that only the solution of Part A of approach 1 ensures an exact solution of the problem, while Part B of approach 1 and Parts A and B of approach 2 are heuristic. Furthermore, note that the objective function (50) (with the breaking tie term), when used in Part A of approach 2 with $\delta_{ik} = l_i$, helps to decrease the total necessary length to pack all boxes from all destinations (see Tables 5 and 6 for instances A_{20}, B_1, B_5, B_{10} and B_{20}). The solutions for instances A_1 in Tables 5 and 6, and for instances B_1 and B_{10} in Table 6, are optimal based on the results of Table 3, and the solutions for instances B_5 and B_{20} in Tables 5 and 6 are better than the ones in Table 2. Applying Part B of approach 1 with $\delta_{ik} = 0$ was necessary for solving instances A_1, A_5 and A_{20} (see Table 4), and applying Part B of approach 2 with $\delta_{ik} = l_i$ was necessary for solving instance A_5 with (32) and (50) (see Tables 5 and 6, respectively). In the later case, it is interesting to note that the objective function (50) was not able to produce better results than the objective function (32) in Part A of approach 2, but it helped Part B of this approach, when applied, to pack a higher volume of boxes.

Table 5 Results obtained with approach 2 with $\delta_{ik} = l_i$ with (32)

	L	$L_1'^*$	$L_2'^*$	$L_3'^*$	$L_n'^*$	Gap (%)	Time (s)	Vol. (%)	No. boxes left out
A_1	15	4	6	10	10	0.000	1.05	100.00	0
A_5	15	5	10	15	15	0.000	248.88	100.00	0
	12	5	10	10	10	0.000	0.21	83.90	7
A_{10}	15	4	8	12	12	20.000	0.48	100.00	0
A_{20}	15	5	9	12	12	20.000	4.90	100.00	0
B_1	15	4	8	11	11	10.000	82.04	100.00	0
B_5	15	4	8	11	11	10.000	12.94	100.00	0
B_{10}	15	4	8	11	11	10.000	5.88	100.00	0
B_{20}	15	4	8	12	12	20.000	318.72	100.00	0

Table 6 Results obtained with approach 2 with $\delta_{ik} = l_i$ with (50)

	L	$L_1'^*$	$L_2'^*$	$L_3'^*$	$L_n'^*$	Gap (%)	Time (s)	Vol. (%)	No. boxes left out
A_1	15	4	6	10	10	0.000	1.40	100.00	0
A_5	15	5	10	15	15	0.000	348.75	100.00	0
	12	5	10	12	12	0.000	0.24	90.00	5
A_{10}	15	4	8	12	12	20.000	1.70	100.00	0
A_{20}	15	5	9	11	11	10.000	3.00	100.00	0
B_1	15	4	7	10	10	0.000	3613.32	100.00	0
B_5	15	4	7	10	10	0.000	18.05	100.00	0
B_{10}	15	4	7	10	10	0.000	5.89	100.00	0
B_{20}	15	4	7	11	11	10.000	3778.74	100.00	0

For the sake of illustration, Figs. 8 and 9 present the loading patterns for instance B_{10} obtained from model (1)–(9) with $\delta_{ik} = 0$ and l_i, and the ones obtained from the two approaches 1 and 2 with $\delta_{ik} = 0$ and l_i (with and without the breaking tie function), respectively. Note that approach 2 with $\delta_{ik} = l_i$ (with the breaking tie function) was able to find the same solution found by model (1)–(9), for the same value of δ_{ik}.

4.2 Results with instances from the literature

We have also performed computational tests with approaches 1 and 2 with $\delta_{ik} = 0$ and l_i (with and without the breaking tie function) and the eight instances from Christensen and Rousøe (2009), based on real-world data from a Danish company distributing construction products. Table 7 shows the number of destinations and boxes for each of these eight instances, the length of the container and the minimum necessary length $L_k'^*$ to pack all boxes from all destinations, the runtime (in seconds) spent to solve each of the instances, the fraction of volume (in %) occupied by the boxes loaded in the container and the number of boxes left out of the container. The container dimensions considered in these computational tests were either $L = 1440$ or $L = 720$ (the study in Christensen and Rousøe 2009 considers

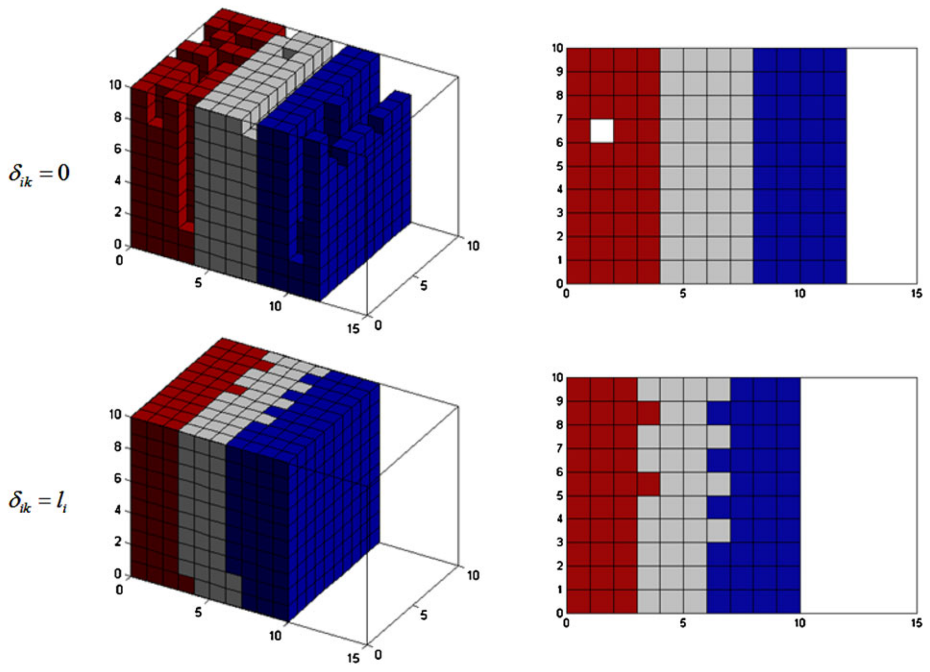


Fig. 8 Solutions of model (1)–(9) with $\delta_{ik} = 0$ and l_i for instance B_{10}

this dimension for the length), $W = 250$ and $H = 280$. We also limited the size of sets X, Y and Z, by excluding the smaller dimensions along each axis, at a time, until no more that 20 positions were set. Therewith, we note that all approaches (including Part A of approach 1) are now heuristic.

Note in Table 7 that only for instance 7, approach 1 with $\delta_{ik} = 0$ was not able to find a feasible solution. For instances 2, 3 and 6, this approach was able to find solutions that packed all boxes inside the given length of the container, while for instances 1, 4, 5 and 8, to pack all boxes a container with a greater length L would be necessary. For solving these instances, Part B of the approach was applied to produce solutions that fit in the container length $L = 720$. It is worth noting that these results were achieved with our most conservative approach, that is, with $\delta_{ik} = 0$. Approach 2 with (32) was unable to find feasible solutions for instances 4, 6 and 8, while Part B of the approach was applied to instances 3, 5 and 7. Approach 2 with (50) was unable to find feasible solutions only for instances 6 and 8, while Part B of the approach was applied to instances 3, 4, 5 and 7. We note that, for instance 5 and approach 2, the use of the objective function (50) had an inverse effect in both Parts A and B of the approach, since it worsened the total necessary length to load all boxes of all destinations (in case of Part A) and the fraction of volume of the boxes loaded (in case of Part B). It should be observed that the approach proposed in Christensen and Rousøe (2009) is not strictly comparable to ours, since their definition of multi-dropping is different from ours. In their approach, a certain box can occupy any empty space inside the container, if there is some access to the box at every drop-off point, regardless of the value of parameter δ_{ik}.

Table 7 Results obtained with approaches 1 and 2 with $\delta_{ik} = 0$ and l_i for the instances of Christensen and Rousøe (2009)

	n	No. boxes	Approach with														
			$\delta_{ik} = 0$					$\delta_{ik} = l_i$ with (32)					$\delta_{ik} = l_i$ with (50)				
			L	$L_n'^*$	Time (s)	Vol. (%)	No. boxes left out	L	$L_n'^*$	Time (s)	Vol. (%)	No. boxes left out	L	$L_n'^*$	Time (s)	Vol. (%)	No. boxes left out
#1	2	3	1440	772	3.84	100.00	0	1440	520	1.43	100.00	0	1440	520	1.60	100.00	0
			720	520	2.84	80.13	1										
#2	1	3	1440	85	0.43	100.00	0	1440	85	0.51	100.00	0	1440	85	0.56	100.00	0
#3	2	9	1440	652	84.73	100.00	0	1440	784	3601.02	100.00	0	1440	784	1884.19	100.00	0
								720	573	0.60	69.36	1	720	573	0.60	69.36	1
#4	3	22	1440	1201	42.74	100.00	0	1440	–	–	–	–	1440	1200	3792.00	100.00	0
			720	480	0.29	52.59	13						720	720	2929.66	69.46	7
#5	4	8	1440	1370	2.39	100.00	0	1440	1220	6.27	100.00	0	1440	1270	6.98	100.00	0
			720	530	0.02	48.77	4	720	530	1.64	58.02	2	720	720	2.30	50.18	2
#6	2	8	1440	450	430.71	100.00	0	1440	–	–	–	–	1440	–	–	–	–
#7	1	11	1440	–	–	–	–	1440	1410	2.21	100.00	0	1440	1410	2.29	100.00	0
								720	680	1.78	71.99	4	720	680	1.81	71.99	4
#8	6	19	1440	1466	0.95	100.00	0	1440	–	–	–	–	1440	–	–	–	–
			720	516	0.38	61.94	12										

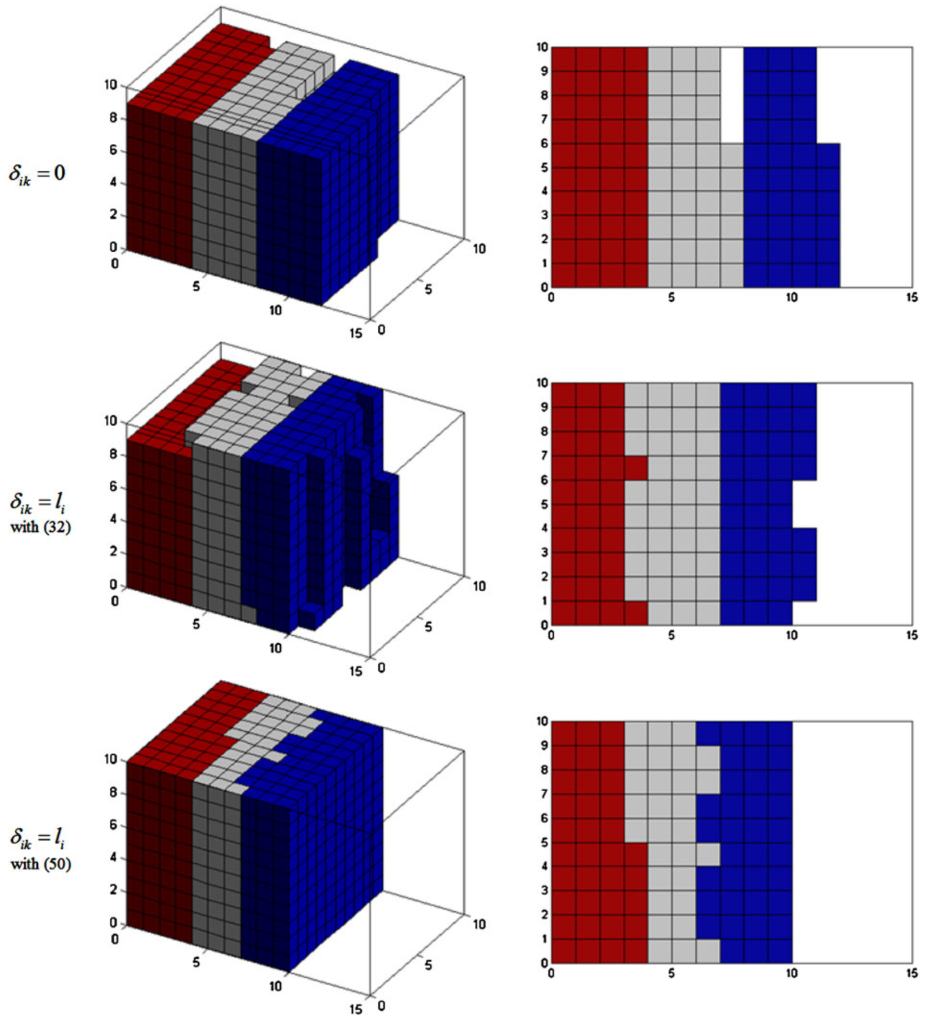


Fig. 9 Solutions of approaches 1 and 2 with $\delta_{ik} = 0$ and l_i (with and without the raking tie function) for instance B_{10}

5 Conclusion

In this paper, we present approaches based on a mixed integer linear programming model for packing problems of rectangular boxes into a container or truck considering multi-dropping, as well as the vertical stability of the cargo. The model and the approaches can be easily extended to take into account other practical considerations, such as horizontal stability of the cargo, load bearing (including fragility) of the boxes and weight limit of the cargo. We assume that the delivery route of the container is already known in advance and that the volume of the cargo of all destinations is less than or equal to the container volume. The objective is to determine the loading pattern that packs the maximum volume (or value) of boxes, taking into account the sequence in which the boxes are unloaded from (or are loaded into) the container, without additional handling requirements when each drop-off point of the route is reached. Computational tests using the proposed model and the approaches were performed with randomly generated instances and instances from the literature, using the GAMS/CPLEX software. The results show that the model and the approaches are consistent and properly represent the situations, although only problems of moderate size can be solved optimally. The proposed model and approaches can be useful to motivate future research in order to solve more realistic container loading problems, so as to deal with the integrated vehicle routing and container loading problem.

Acknowledgements The authors would like to thank the two anonymous reviewers for their useful comments and suggestions. This research was partially supported by FAPESP (grants 06/03496-3, 07/00209-6 and 09/07423-9) and CNPq (grant 522973/95-7).

References

Beasley, J. E. (1985). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, *33*(1), 49–64.

Bischoff, E. E., & Marriott, M. D. (1990). A comparative evaluation of heuristics for container loading. *European Journal of Operational Research*, *44*(2), 267–276.

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

Christensen, S. G., & Rousøe, D. M. (2009). Container loading with multi-drop constraints. *International Transactions in Operational Research*, *16*(6), 727–743.

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

Dowsland, W. B. (1991). Three-dimensional packing—solution approaches and heuristic development. *International Journal of Production Research*, *29*(8), 1673–1685.

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

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, *201*(3), 751–759.

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, *40*(3), 342–350.

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

Haessler, R. W., & Talbot, F. B. (1990). Load planning for shipments of low density products. *European Journal of Operational Research*, *44*(2), 289–299.

Han, C. P., Knott, K., & Egbelu, P. J. (1989). A heuristic approach to the three-dimensional cargo-loading problem. *International Journal of Production Research*, *27*(5), 757–774.

Iori, M., Gonzalez, J. S., & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, *41*(2), 253–264.

Jin, Z., Ohno, K., & Du, J. (2004). An efficient approach for the three-dimensional container packing problem with practical constraints. *Asia-Pacific Journal of Operational Research*, *21*(3), 279–295.

Junqueira, L., Morabito, R., & Yamashita, D. S. (2012). Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers and Operations Research*, *31*(1), 74–85.

Lai, K. K., Xue, J., & Xu, B. (1998). Container packing in a multi-customer delivering operation. *Computers & Industrial Engineering*, *35*(1–2), 323–326.

Lin, J. L., Chang, C. H., & Yang, J. Y. (2006). A study of optimal system for multiple-constraint multiple-container packing problems. In *Proceedings of the 19th international conference on industrial, engineering and other applications of applied intelligent systems*, Annecy (Vol. 4031, pp. 1200–1210).

Lins, L., Lins, S., & Morabito, R. (2002). An n-tet graph approach for non-guillotine packing of n-dimensional boxes into an n-container. *European Journal of Operational Research*, *141*(2), 421–439.

Miyazawa, F. K., & Wakabayashi, Y. (1999). Approximation algorithms for the orthogonal Z-oriented three-dimensional packing problem. *SIAM Journal on Computing*, *29*(3), 1008–1029.

Morabito, R., & Arenales, M. (1994). An and/or-graph approach to the container loading problem. *International Transactions in Operational Research*, *1*(1), 59–73.

Moura, A., & Bortfeldt, A. (2009). A packing and routing application for a Portuguese trading company. In *Book of abstracts of the 6th ESICUP meeting*, Valencia.

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

Moura, A., & Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems. *OR-Spektrum*, *31*(4), 775–800.

Parreño, F., Alvarez-Valdes, R., Oliveira, J. F., & Tamarit, J. M. (2010). A hybrid GRASP/VND algorithm for two- and three-dimensional bin packing. *Annals of Operation Research*, *179*(1), 203–220.

Scheithauer, G., Terno, J., Riehme, J., & Sommerweiss, U. (1996). *A new heuristic approach for solving the multi-pallet packing problem*. Technical report, MATH-NM-03-1996. Technische Universität Dresden, Dresden.

Silva, J. L. C., Soma, N. Y., & Maculan, N. (2003). A greedy search for the three-dimensional bin packing problem: the packing static stability case. *International Transactions in Operational Research*, *10*(2), 141–153.

Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems*, *10*(2), 255–271.

Terno, J., Scheithauer, G., Sommerweiss, U., & Riehme, J. (2000). An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research*, *123*(2), 372–381.

Tetris (2009). *25th anniversary*. Available at: http://www.tetris.com. Accessed in: 14 out 2009.