WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 23 (2016) 187–213
DOI: 10.1111/itor.12111

# A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application

C. Paquay, M. Schyns and S. Limbourg

*HEC Management School, QuantOM, University of Liege (ULg), Liège, Belgium*
*E-mail: cpaquay@ulg.ac.be [Paquay]; M.Schyns@ulg.ac.be [Schyns]; Sabine.Limbourg@ulg.ac.be [Limbourg]*

Received 31 May 2013; received in revised form 9 May 2014; accepted 30 May 2014

**Abstract**

The present paper discusses the problem of optimizing the loading of boxes into containers. The goal is to minimize the unused volume. This type of problem belongs to the family of multiple bin size bin packing problems (MBSBPP). The approach includes an extensive set of constraints encountered in real-world applications in the three-dimensional case: the stability, the fragility of the items, the weight distribution, and the possibility to rotate the boxes. It also includes the specific situation in which containers are truncated parallelepipeds. This is typical in the field of air transportation. While most papers on cutting and packing problems describe *ad hoc* procedures, this paper proposes a mixed integer linear program. The validity of this model is tested on small instances.

*Keywords:* packing problems; loading problems; linear programming; air transport; weight distribution

## 1. Introduction

The aim of this paper is to develop a mathematical linear model for the packing problem of a strongly heterogeneous assortment of boxes into a selection of containers of various shapes. The objective is to minimize the unused volume.

This problem belongs to the family of cutting and packing problems. We can label it as a three-dimensional (3D) multiple bin size bin packing problem (MBSBPP) using the typology defined by Wäscher et al. (2007). Indeed, this is an input minimization problem for which the dimensions of all objects are fixed, the small items being strongly heterogeneous, and the assortment of large objects, such as the containers, is weakly heterogeneous. Because our problem is a packing problem in three dimensions, therefore it also belongs to the family of container loading problems according to the definition given in Bortfeldt and Wäscher (2013).

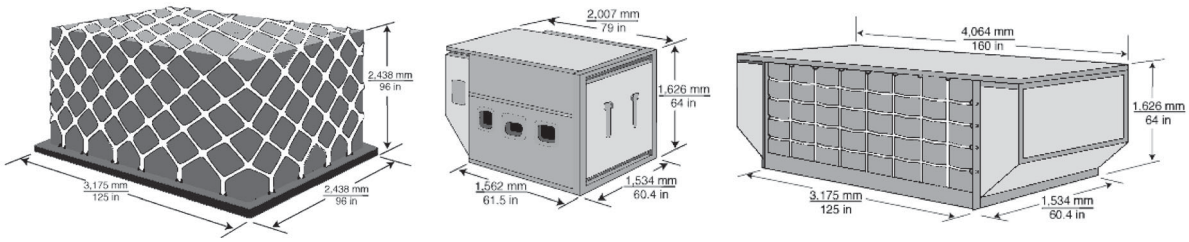


Fig. 1. Different shapes of ULDs.

In this work, we extend the definition of the MBSBPP to include the situations in which the large objects may be truncated parallelepipeds. This is of particular importance in the field of air transportation. In this context, containers are called unit load devices (ULDs). A ULD is an assembly of components consisting of a container or of a pallet covered with a net, so as to provide standardized size units for individual pieces of baggage or cargo, and to allow for rapid loading and unloading (Limbourg et al., 2012). ULDs may have specific shapes to fit inside an aircraft. Several common ULDs are illustrated in Fig. 1.

Papers on cutting and packing problems start by defining geometric constraints to ensure that the small items lie entirely and without overlap inside the large objects. Bortfeldt and Wäscher (2013), on the basis of the paper by Bischoff and Ratcliff (1995), present a broad set of additional constraints that might be encountered in practical packing situations. Of these, our model takes the following into account: the container weight limit, the orientation constraints, the load stability, the load-bearing strength or fragility of items, and the weight distribution within a container. These constraints are explained in more details in Section 2. Some of the remaining constraints are not relevant in our situation; for example, shipment priorities and multidrop situations. We indeed assume that all the boxes must be loaded and grouped by destination, as it is typically the case with ULDs. As discussed, we also take into account the specific shape of the containers.

Our approach differs from the existing works not only by the set of constraints integrated into the model, but also by other factors such as the type of problem, its representation, its method of resolution, and its dimensionality. Some related problems do not correspond to MBSBPP but fall in other categories of Wäscher et al.'s (2007) typology. The 3D single bin size bin packing problem (SBSBPP) arises when only one type of container is allowed. The multiple stock size cutting stock problem (MSSCSP) differs from the MBSBPP by a weakly heterogeneous assortment of small items. The single knapsack problem (SKP) is an output maximization problem. In this case, the goal is to select items, from within a strongly heterogeneous set of small items, to pack inside a single large object. We label it as a single large object placement problem (SLOPP) when the set of small items is weakly heterogeneous.

Several papers treated the problems in two dimensions either with exact approaches as in Herz (1972), Beasley (1985), and Hadjiconstantinou and Christofides (1995), or with heuristics as in Christofides and Whitlock (1977) for the SLOPP. More recently, exact approaches have been developed for the 3D SKP (Tsai et al., 1993; Padberg, 2000; Junqueira et al., 2012), but few for the MBSBPP (Chen et al., 1995; Westerlund et al., 2005). Most of the papers about 3D MBSBPP present heuristics (Martello et al., 2000; Terno et al., 2000; Jin et al., 2003; Techanitisawad and

Tangwiwatwong, 2004; Brunetta and Gregoire, 2005; Lin et al., 2006; Almeida and Figueiredo, 2010; Ceschia and Schaerf, 2013). In the papers cited previously, only those presenting exact methods provide a mathematical formulation. Chen et al. (1995) and Westerlund et al. (2005) are the only studies to consider mathematical formulations for the MBSBPP. However, some parts of other SKP models could also be reused in this context. The main features of the papers cited previously are summarized in Table 1.

Our main aim is to provide a rich and realistic mathematical representation of this extended MBSBPP. To the best of our knowledge, we are the first to propose a linear mathematical formulation taking into account all the mentioned constraints. Moreover, we test the validity of this linear mathematical model on small instances. These results can provide some insights into how to use this formulation for building new heuristics.

This paper is organized as follows. In Section 2, the MBSBPP and the constraints considered within our scope are briefly described. The mathematical model is depicted in Section 3. In Section 4, some comments on the implementation and results are discussed. Finally, conclusions are drawn in Section 5 as well as some perspectives about the future research.

## 2. Problem definition

The aim of this paper is to give a mathematical formulation for the following problem:

| | |
|---|---|
| min | unused space |
| s.t. | each box assigned to exactly one used bin |
| | each box within the limits of the bin |
| | the total weight of the boxes inside a ULD $\leq$ maximum capacity |
| | orthogonal placement |
| | no overlap |
| | orientation constraints |
| | special shape of the ULDS |
| | stability |
| | fragility |
| | weight distribution. |

The boxes are assumed to be placed orthogonally, that is, the edges of the boxes have to be either parallel or perpendicular to those of the containers. In practice, however, some boxes may not rotate in all directions because of their content; for example, some products may not turn upside-down (Chen et al., 1995; Terno et al., 2000; Chan et al., 2006; Almeida and Figueiredo, 2010; Junqueira et al., 2012; Ceschia and Schaerf, 2013). These constraints are called "orientation constraints."

As mentioned above, some ULDs may have a special shape to fit in the fuselage of the aircraft: some of them look like parallelepipeds that have been cut as shown in Fig. 2. On the left-hand side of Fig. 2, an aircraft cross-section where three ULDs are represented is shown; two of them are on the lower deck and the third one is on the upper deck. On the right-hand side, two ULDs are loaded on the lower deck of a Boeing 747 aircraft.

Table 1
Constraints considered in publications on packing problems

| Publications | Type | Approach | Weight limit | Weight distribution | Orientation | Fragility | Stability | Shape |
|---|---|---|---|---|---|---|---|---|
| Tsai et al. (1993) | SKP | Exact | | | | | | |
| Padberg (2000) | SKP | Exact | | | | | | |
| Junqueira et al. (2012) | SKP | Exact | | | × | × | × | |
| Chen et al. (1995) | MBSBPP | Exact | | × | × | | | |
| Martello et al. (2000) | SBSBPP | Heuristics | | | × | | | |
| Terno et al. (2000) | MBSBPP | Heuristics | × | | × | × | × | |
| Jin et al. (2003) | MBSBPP | Heuristics | | | × | | × | |
| Techanitisawad and Tangwiwatwong (2004) | MBSBPP | Heuristics | | × | × | × | × | |
| Westerlund et al. (2005) | MBSBPP | Exact | | | | | | |
| Lin et al. (2006) | MBSBPP | Heuristics | | | × | × | × | |
| Chan et al. (2006) | MBSBPP | Heuristics | × | × | × | | × | × |
| Almeida and Figueiredo (2010) | MBSBPP | Heuristics | | | × | | | |
| Ceschia and Schaerf (2013) | MBSBPP | Heuristics | × | | × | × | × | |

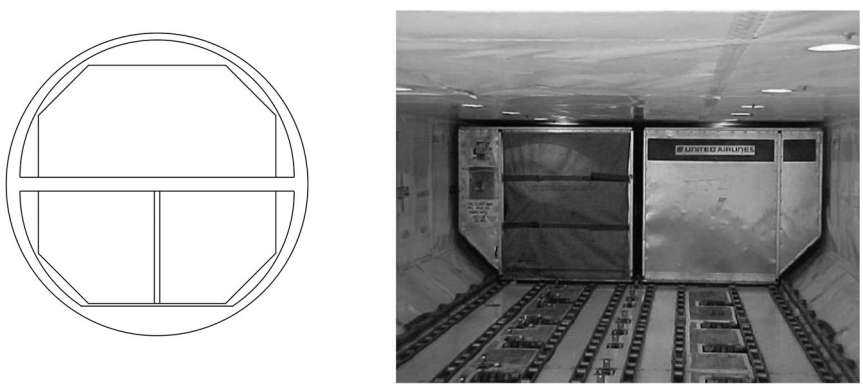


Fig. 2. Special shapes of some ULDs to fit in the fuselage.

On the aircraft cross-section of Fig. 2, these special ULDs look like rectangles from which one (or more) corner has been cut. For example, the bins with a shape similar to the two ULDS on the left-hand side of the Fig. 3 can fit on the lower deck, whereas the two ULDs on the right-hand side can fit on the upper deck. It is obvious that the boxes must lie within the containers and therefore we should pay attention to their particular shapes (e.g., Chan et al., 2006).

Cargo stability involves the vertical (or static) and the horizontal (or dynamic) stability (Terno et al., 2000; Jin et al., 2003; Techanitisawad and Tangwiwatwong, 2004; Chan et al., 2006; Lin et al., 2006; Junqueira et al., 2012; Ceschia and Schaerf, 2013). For the sake of vertical stability, the bottom side of each box needs to be supported by the top face of other boxes, by a cut in the case of a special shape or by the container floor. This constraint is also called "static stability" as it deals

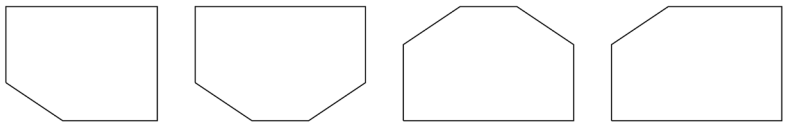

Fig. 3. Special shapes of ULDS (projection on the $XZ$ plane).

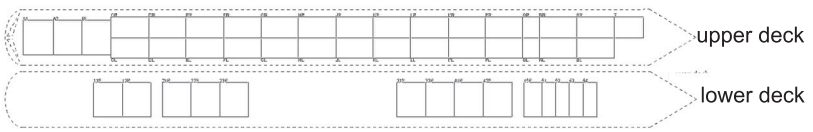


Fig. 4. Possible position for the ULDs inside a Boeing 747.

with static containers. The vertical stability excludes floating boxes. The horizontal stability refers to the capacity of the box to withstand the inertia of its own body when being moved. The boxes remain in their position with respect to $x$ and $y$ axes, hence the name of "horizontal stability." This paper only considers vertical stability because horizontal stability could be obtained by adding a special sheet increasing the friction coefficient or by bounding unstable boxes.

Load-bearing strength refers to the maximum number of boxes that can be stacked up. More generally, it refers to the maximum pressure that can be applied over the top face of a box without damaging it. Different strategies have been developed to manage this feature of the cargo (Terno et al., 2000; Junqueira et al., 2012; Ceschia and Schaerf, 2013). For instance, a box is said to be fragile if no box can be placed on its top face. This constraint is quite important in practice because it prevents damaging products contained in a fragile box. This paper takes box fragility into account.

Once filled, these ULDs are loaded into a compartmentalized cargo aircraft with some technical and safety constraints. The structure of the Boeing 747 is shown in Fig. 4. Each rectangle represents a possible position for several ULDs.

The ULDs are loaded in such a way that the center of gravity (CG) of the loaded plane is as close as possible to a recommended position determined by safety and fuel economy considerations (Amiouny et al., 1992; Mongeau and Bès, 2003; Fok and Chun, 2004; Limbourg et al., 2012; Vancroonenburg et al., 2014). According to the control and loading manual of some airline companies such as Boeing (Boeing, 2006), the CG of each ULD should lie in a determined area around the geometrical center of the ULD. Based on this idea, to calculate the CG of the plane and to ensure some weight constraints, the CG of each ULD is considered as a point in the center of the position occupied. This implies some uniformity about the weight distribution inside the ULDs. Therefore, this assumption is integrated as a constraint in the model as in other studies (Chen et al., 1995; Gehring and Bortfeldt, 1997; Davies and Bischoff, 1999; Techanitisawad and Tangwiwatwong, 2004; Chan et al., 2006; Kaluzny and Shaw, 2009; Baldi et al., 2012; Moon and Nguyen, 2013). Moreover, ULDs have to be loaded in such a way that a limit weight is satisfied at each slice of an inch of the aircraft. Satisfying the assumption of uniform weight distribution inside a ULD is therefore crucial.

Bortfeldt and Wäscher (2013) provide more details and literature on the MBSBPP.

## 3. Mathematical formulation

The description of the problem is as follows. A set of $n$ rectangular boxes of dimensions $l_i \times w_i \times h_i$ and weight $m_i$ ($i \in \{1, \ldots, n\}$) has to be packed into $m$ available ULDs of dimensions $L_j \times W_j \times H_j$, a maximal capacity, also called maximum gross weight, $C_j$, and a volume $V_j$ while minimizing the unused volume. The packing has to satisfy different geometric and specific constraints that will be specified later. This paper proposes a mathematical model for packing the boxes into the ULDs.

### 3.1. Parameters

The following data are known: the number of boxes to be packed; dimensions and weight of each box; and dimensions, maximum gross weight, and volume of each container. We assume that all these numbers are integers, even if this means changing the scale. These parameters are defined as

| | | |
|---|---|---|
| $n$ | Total number of boxes to be packed, | |
| $m$ | Total number of available ULDs, | |
| $l_i \times w_i \times h_i$ | Length × width × height of box $i$, | $\forall i$, |
| $m_i$ | Weight of box $i$, | $\forall i$, |
| $L_j \times W_j \times H_j$ | Length × width × height of container $j$, | $\forall j$, |
| $C_j$ | Maximum gross weight of container $j$, | $\forall j$, |
| $V_j$ | Volume of container $j$, | $\forall j$, |

$i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$.

To ensure at least one feasible solution, some conditions are assumed to be satisfied: for example, the weight of each box is supposed to be less than or equal to the maximum capacity of the bins: $m_i \leq \max_j C_j \, \forall i \in \{1, \ldots, n\}$.

### 3.2. Variables

Let us represent the situation in a 3D geometric space. Without loss of generality, the axes of the coordinate system are assumed to be placed so that the length $L_j$ (resp. width $W_j$, height $H_j$) of the container $j$ lies on the $x$-axis (resp. $y$-axis, $z$-axis) $\forall j \in \{1, \ldots, m\}$. The origin of this coordinate system lies on the front left bottom corner of the containers. A representation is given in Fig. 5.

Here are the various variables used in the model. Note that the subscripts relate to indices and superscripts relate to fixed objects.

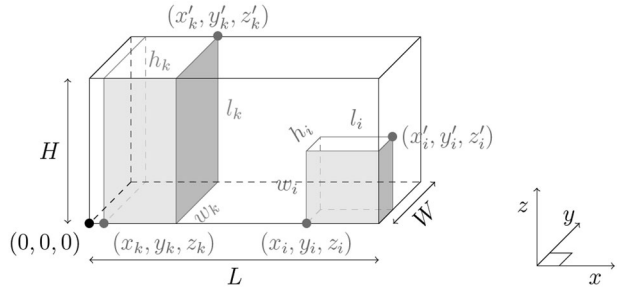


Fig. 5. Representation of some parameters and variables: the container is shown as black lines, boxes $i$ and $k$ are shown as grey, and the coordinate system is on the right.

$$p_{ij} = \begin{cases} 1 & \text{if box } i \text{ is in container } j, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i, j,$$

$$u_j = \begin{cases} 1 & \text{if container } j \text{ is used,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall j,$$

$(x_i, y_i, z_i)$ location of the front left bottom corner of box $i$, $\forall i$,

$(x'_i, y'_i, z'_i)$ location of the rear right top corner of box $i$, $\forall i$,

$$r_{iab} = \begin{cases} 1 & \text{if the side } b \text{ of box } i \text{ is along the } a\text{-axis,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i,$$

$$x^p_{ik} = \begin{cases} 1 & \text{if box } i \text{ is on the right of box } k (x'_k \le x_i), \\ 0 & \text{otherwise} (x_i < x'_k), \end{cases} \qquad \forall i,$$

$$y^p_{ik} = \begin{cases} 1 & \text{if box } i \text{ is behind box } k (y'_k \le y_i), \\ 0 & \text{otherwise } (y_i < y'_k), \end{cases} \qquad \forall i,$$

$$z^p_{ik} = \begin{cases} 1 & \text{if box } i \text{ is above } k (z'_k \le z_i), \\ 0 & \text{otherwise } (z_i < z'_k), \end{cases} \qquad \forall i,$$

$i, k \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$, $a, b \in \{1, 2, 3\}$.

Note that the variables $(x_i, y_i, z_i)$ and $(x'_i, y'_i, z'_i)$ are also assumed to be integers. These variables describe the position of the box $i$ inside the container. They are represented in Fig. 5.

Since the boxes can rotate orthogonally, the variables $r_{iab}$ are introduced to describe the orientation of the box $i$ inside the container. The index $b$ indicates the side of the box, that is $b \in \{l := 1, w := 2, h := 3\}$, whereas $a$ indicates the axis, that is $a \in \{x := 1, y := 2, z := 3\}$. They specify which side of the box $i$ is along which axis. For example, these variables are equal to

$$\begin{array}{llllll} r_{i11} = 1 & r_{i12} = 0 & r_{i13} = 0 & r_{j11} = 0 & r_{j12} = 0 & r_{j13} = 1 \\ r_{i21} = 0 & r_{i22} = 0 & r_{i23} = 1 & r_{j21} = 0 & r_{j22} = 1 & r_{j23} = 0 \\ r_{i31} = 0 & r_{i32} = 1 & r_{i33} = 0 & r_{j31} = 1 & r_{j32} = 0 & r_{j33} = 0 \end{array}$$

in Fig. 5.

To ensure that there is no overlap, we need to know the relative position of two boxes. Thus, the variable $x^p_{ik}$ (resp. $y^p_{ik}$, $z^p_{ik}$) is equal to 1 if the box $i$ is on the right side (resp. behind, above) of box $k$. These variables describe all the situations. Indeed, for instance, if the box $i$ is on the left side of box $k$, it means that box $k$ is on the right side of box $i$, and then $x^p_{ki} = 1$.

In Fig. 5, one has

| $x^p$ | $i$ | $k$ |
|---|---|---|
| $i$ | 0 | 1 |
| $k$ | 0 | 0 |

| $y^p$ | $i$ | $k$ |
|---|---|---|
| $i$ | 0 | 0 |
| $k$ | 0 | 0 |

Even if the definition of the $z^p_{ik}$ is the same as $x^p_{ik}$ and $y^p_{ik}$, we will see in Section 3.5 that they are not fully determined because it is not necessary. Indeed, only half of the definition will be guaranteed by the constraints: if $z^p_{ik} = 1$, then we are sure that $z'_k \leq z_i$. On the contrary, if $z^p_{ik} = 0$, then we have no information.

### 3.3. Objective function

The objective function consists in minimizing the unused volume of the ULDs

$$\sum_{j=1}^{m} u_j V_j - \sum_{i=1}^{n} l_i w_i h_i. \tag{1}$$

Since $l_i$, $w_i$, $h_i$ are parameters that are initially given, the term $\sum_{i=1}^{n} l_i w_i h_i$ is a constant. Therefore, the volume of the used containers is minimized:

$$\sum_{j=1}^{m} u_j V_j. \tag{2}$$

If the parameters $V_j$, $\forall j$ represented the cost of a ULD, then the objective function would become a cost minimization.

### 3.4. Constraints

According to Bortfeldt and Wäscher (2013), some constraints are considered to be basic constraints (the geometric ones) and the others to be specific ones. The fact that each box lies within exactly one ULD and does not overlap represents the geometric constraints. The specific constraints include the orientation constraints, special shapes of the ULDs, stability, fragility, and even weight distribution.

Before going into any more detail, some parameters are introduced as follows:

$$L = \max_{j\in\{1,\ldots,m\}} L_j, \qquad W = \max_{j\in\{1,\ldots,m\}} W_j, \qquad H = \max_{j\in\{1,\ldots,m\}} H_j.$$

*3.5. Geometric constraints*

Here are the geometric constraints of the model:

$$\sum_{i=1}^{n} m_i p_{ij} \leq C_j u_j, \qquad \forall j, \tag{3}$$

$$\sum_{j=1}^{m} p_{ij} = 1, \qquad \forall i, \tag{4}$$

$$x'_i \leq \sum_{j=1}^{m} L_j p_{ij}, \qquad \forall i, \tag{5}$$

$$y'_i \leq \sum_{j=1}^{m} W_j p_{ij}, \qquad \forall i, \tag{6}$$

$$z'_i \leq \sum_{j=1}^{m} H_j p_{ij}, \qquad \forall i, \tag{7}$$

$$x'_i - x_i = r_{i11} l_i + r_{i12} w_i + r_{i13} h_i, \qquad \forall i, \tag{8}$$

$$y'_i - y_i = r_{i21} l_i + r_{i22} w_i + r_{i23} h_i, \qquad \forall i, \tag{9}$$

$$z'_i - z_i = r_{i31} l_i + r_{i32} w_i + r_{i33} h_i, \qquad \forall i, \tag{10}$$

$$\sum_{a} r_{iab} = 1, \qquad \forall i, b, \tag{11}$$

$$\sum_{b} r_{iab} = 1, \qquad \forall i, a, \tag{12}$$

$i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$, $a, b \in \{1, 2, 3\}$.

The maximum capacity of each container $j$ cannot be exceeded, which is ensured by constraints (3). This set of constraints, in conjunction with the minimization of the objective function, fully determines the values of the variables $u_j$. Constraints (4) verify that each box is allocated to exactly one container. Constraints (5)–(7) ensure that the boxes do not exceed their container size. Constraints (8)–(12) describe that the boxes can rotate orthogonally in the container. Note that (8)–(10) imply $x_i < x'_i$, $y_i < y'_i$, $z_i < z'_i$.

The following constraints ensure that there is no overlap, that is, two boxes cannot occupy a same portion of the space:

$$x^p_{ik} + x^p_{ki} + y^p_{ik} + y^p_{ki} + z^p_{ik} + z^p_{ki} \geq (p_{ij} + p_{kj}) - 1, \qquad \forall i, k, j, \tag{13}$$

$$x'_k \leq x_i + (1 - x^p_{ik})L, \qquad \forall i, k, \tag{14}$$

$$x_i + 1 \leq x'_k + x^p_{ik} L, \qquad \forall i, k, \tag{15}$$

$$y'_k \leq y_i + (1 - y^p_{ik})W, \qquad \forall i, k, \tag{16}$$

$$y_i + 1 \leq y'_k + y^p_{ik} W, \qquad \forall i, k, \tag{17}$$

$$z'_k \leq z_i + (1 - z^p_{ik}) H, \qquad \forall i, k, \tag{18}$$

$i, k \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$.

When the variables $x^p_{ik}$, $x^p_{ki}$, $y^p_{ik}$, $y^p_{ki}$, $z^p_{ik}$, or $z^p_{ki}$ equal 1, the two boxes $i$ and $k$ do not overlap along any of the axes. To prevent having two boxes occupying a same portion of space, it is sufficient to allow no overlap along at least one of the axes, that is, at least one of these variables must equal 1. It leads to constraints (13). An overlap can happen only if two boxes are in the same bin, which is expressed by the right-hand side of constraints (13). We have not fully determined the variables $x^p_{ik}$, $y^p_{ik}$, and $z^p_{ik}$ so far. It is not required to manage the geometric constraints. However, a full determination of $x^p_{ik}$ and $y^p_{ik}$ will be necessary for handling some specific constraints. For this reason, constraints (14) and (15) are added to the model. These constraints ensure that $x^p_{ik} = 1$ (resp. $y^p_{ik} = 1$), if and only if $x_i \geq x'_k$ (resp. $y_i \geq y'_k$). Note that the parameters $L$, $W$, $H$ are used in these constraints because we do not know in which container the boxes $i$ and $k$ lie.

### *3.6. Specific constraints*

As mentioned above, applying the MBSBPP to the real-world situations implies some specific constraints.

*Orientation constraints.* Some boxes may not rotate in all directions because of their contents; for instance, some products may not turn upside down. For this purpose, some new parameters are introduced for each box $i$, $i \in \{1, \ldots, n\}$:

$$l^+_i = \begin{cases} 1 & \text{if the length of box } i \text{ could be in a vertical position,} \\ 0 & \text{otherwise,} \end{cases}$$

$$w^+_i = \begin{cases} 1 & \text{if the width of box } i \text{ could be in a vertical position,} \\ 0 & \text{otherwise,} \end{cases}$$

$$h^+_i = \begin{cases} 1 & \text{if the height of box } i \text{ could be in a vertical position,} \\ 0 & \text{otherwise.} \end{cases}$$

If all these parameters are set to one, each box is free to rotate in any direction. More precisely, in this case constraints (8)–(12) allow six orientations for each box. If one parameter is set to zero, the goal is to forbid two of these six configurations. For example, only the four configurations depicted in Fig. 6 should remain feasible with $l^+_i$ set to 0.

Likewise, if two parameters equal 0, only two configurations of the box remain possible. Unless at least one of these parameters equals one, there is no possible configuration. Keeping in mind that the variables $r_{i3b}$ describe which side of the box $i$ is along the $z$-axis—that is, determining the value of $z'_i$—constraints (19)–(21) come naturally

$$r_{i31} \leq l^+_i, \qquad \forall i, \tag{19}$$

$$r_{i32} \leq w^+_i, \qquad \forall i, \tag{20}$$

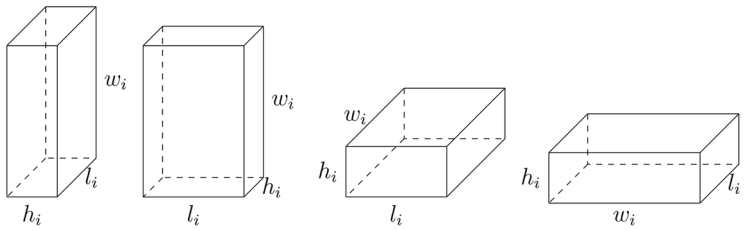


Fig. 6. Possible configurations for the box $i$ if the length of the box could not be along the $z$-axis.

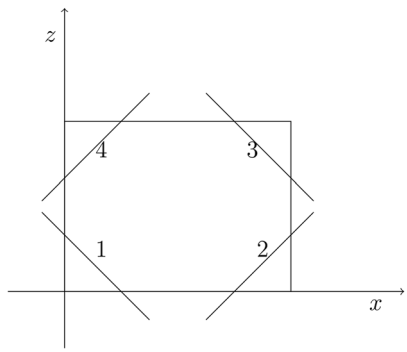


Fig. 7. Possible cuts (projection on the $XZ$ plane).

$$r_{i33} \leq h_i^+, \qquad \forall i, \tag{21}$$

$i \in \{1, \ldots, n\}$.

*Special shapes of the ULDs.* As explained in the Introduction, some ULDs may have a special shape to fit into the fuselage of the aircraft. There exist four possible cuts for a container. Each cut can be described by the linear equation $z = ax + b$, where $a, b \in \mathbb{R}$, and the left bottom corner of the container before truncation lies at the origin of the coordinate system as shown in Fig. 7. To describe these cuts, eight new parameters (two for each cut $\sigma$, $\sigma \in \{1, \ldots, 4\}$) are added for each bin $j$, $j \in \{1, \ldots, n\}$: $a_{\sigma j}$, $b_{\sigma j} \in \mathbb{R}^+$ as shown in Fig. 7.

To lie inside these types of ULDs, each box has to satisfy constraints associated to the cuts of the bin it is put in:

$$\text{cut 1: } z_i + a_{1j}x_i \geq b_{1j} - M(1 - p_{ij}), \qquad \forall i, j, \tag{22}$$

$$\text{cut 2: } z_i - a_{2j}x'_i \geq -b_{2j} - M(1 - p_{ij}), \qquad \forall i, j, \tag{23}$$

$$\text{cut 3: } z'_i + a_{3j}x'_i \leq b_{3j} + M(1 - p_{ij}), \qquad \forall i, j, \tag{24}$$

$$\text{cut 4: } z'_i - a_{4j}x_i \leq b_{4j} + M(1 - p_{ij}), \qquad \forall i, j, \tag{25}$$

$i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$, where $M = b_1 + H + \max\{a_2, a_3, 1\}L$. If a ULD $j$ has not cut $\sigma$, $\sigma \in \{1, \ldots 4\}$, then the corresponding constraints are deleted from the model before the optimization.

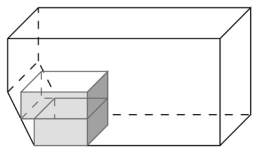

Fig. 8. A box supported by another box and a cut.

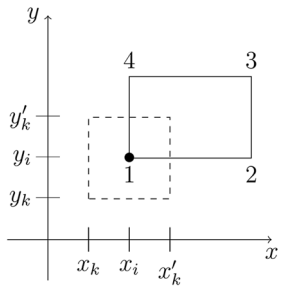


Fig. 9. The vertex $(x_i, y_i)$ from box $i$ (in solid lines) is supported by the box $k$ (in dashed lines) (projection on the $XY$ plane).

Note that cuts 1 and 2 will influence the vertical stability since a box can be supported by these cuts as shown in Fig. 8.

*Vertical stability.* As explained in Section 2, the bottom face of each box has to be supported by the top face of other boxes, by a cut or by the container floor. Thus, the boxes are not displaced with respect to $z$-axis. Especially, the vertical stability excludes floating boxes.

In this model, we do more than just verify if a box is supported. Physically speaking, an object is stable if its CG lies within its support basis. The weight inside the boxes is assumed to be uniform, therefore the CG corresponds to the geometric center of the box. By definition, the support basis is the convex hull of all the contact points. The stability constraints of this model rely on this idea: if a box is not on the ground, then at least three vertices of its basis must be supported. In this way, the CG will always lie in the support basis. We chose a layer reasoning: when a box is added, we ensure that this is stable with respect to the layer below, we do not take into account the boxes that can be stacked above. This is a simplified assumption and in some rare situations, an unstable configuration could arise. To avoid these situations, the model could be easily adapted to ensure that the four vertices of the basis are supported. However, it would imply that the constraints become stronger and some stable configurations based on only three vertices would be rejected. To achieve the stability, with the help of some variables we determine whether a box is on the ground and whether a vertex of the bottom face is correctly supported. A vertex of the bottom face of a box $i$ is correctly supported if there is another box $k$ that has the suitable height to support the vertex, that is, the coordinate along the $z$-axis of the top face of box $k$ equals the coordinate along the $z$-axis of the bottom face of box $i$ ($z'_k = z_i$), and with a particular overlap of the projections of these two boxes on the $XY$ plane. To define this overlap, we consider that the vertices of box $i$ are assigned to a number in the following way: $(x_i, y_i) := 1$, $(x'_i, y_i) := 2$, $(x'_i, y'_i) := 3$, and $(x_i, y'_i) := 4$. As shown in Fig. 9, the vertex 1 (resp. 2, 3, 4) is supported if there exists a box $k$ in the same bin, with the suitable height, such that

$$x_k \leq x_i < x'_k \quad \text{and} \quad y_k \leq y_i < y'_k \qquad \begin{array}{l}(\text{resp. } x_k < x'_i \leq x'_k \quad \text{and} \quad y_k \leq y_i < y'_k,\\ \qquad\quad x_k < x'_i \leq x'_k \quad \text{and} \quad y_k < y'_i \leq y'_k,\\ \qquad\quad x_k \leq x_i < x'_k \quad \text{and} \quad y_k < y'_i \leq y'_k).\end{array}$$

For this purpose, new variables are introduced:

$$g_i = \begin{cases} 1 & \text{if box } i \text{ is on the ground } (z_i = 0),\\ 0 & \text{otherwise,}\end{cases} \qquad \forall i,$$

$$h_{ik} = \begin{cases} 0 & \text{if box } k \text{ has the suitable height to support box } i \ (z_i = z'_k),\\ 1 & \text{otherwise,}\end{cases} \qquad \forall i, k,$$

$$o_{ik} = \begin{cases} 0 & \text{if the projections on the } XY \text{ plane of the boxes}\\ & i \text{ and } k \text{ have a nonempty intersection,}\\ 1 & \text{otherwise,}\end{cases} \qquad \forall i, k,$$

$$s_{ik} = \begin{cases} 1 & \text{if box } k \text{ supports box } i \text{ and are in the same bin,}\\ 0 & \text{otherwise,}\end{cases} \qquad \forall i, k,$$

$$\eta^1_{ik} = \begin{cases} 0 & \text{if } x_k \leq x_i,\\ 1 & \text{otherwise,}\end{cases} \qquad \forall i, k,$$

$$\eta^2_{ik} = \begin{cases} 0 & \text{if } y_k \leq y_i,\\ 1 & \text{otherwise,}\end{cases} \qquad \forall i, k,$$

$$\eta^3_{ik} = \begin{cases} 0 & \text{if } x'_i \leq x'_k,\\ 1 & \text{otherwise,}\end{cases} \qquad \forall i, k,$$

$$\eta^4_{ik} = \begin{cases} 0 & \text{if } y'_i \leq y'_k,\\ 1 & \text{otherwise,}\end{cases} \qquad \forall i, k,$$

$$\beta^l_{ik} = \begin{cases} 1 & \text{if the vertex } l \text{ is supported by box } k,\\ 0 & \text{otherwise,}\end{cases} \qquad \forall i, k, l,$$

$$\gamma^1_i = \begin{cases} 1 & \text{if the box } i \text{ lays on cut 1 of the bin in which it lies,}\\ 0 & \text{otherwise,}\end{cases} \qquad \forall i,$$

$$\gamma^2_i = \begin{cases} 1 & \text{if the box i lays on cut 2 of the bin in which it lies,}\\ 0 & \text{otherwise,}\end{cases} \qquad \forall i,$$

$i, k \in \{1, \ldots, n\}$, $l \in \{1, \ldots, 4\}$.

Variables $s_{ik}$ and $o_{ik}$ seem unnecessary at first sight, but they will be useful for the fragility constraints hereunder.

As developed previously, stability constraints could be written as follows:

$$\sum_{l=1}^{4}\sum_{k=1}^{n}\beta_{ik}^{l}+2\gamma_{i}^{1}+2\gamma_{i}^{2}\geq 3(1-g_{i})\quad\forall i\in\{1,\ldots,n\}. \tag{26}$$

If the box $i$ is not on the ground, constraints (26) ensure that at least three vertices are supported, either by another box $k$, or by a cut. Indeed, if $\sum_{k=1}^{n}\beta_{ik}^{l}=1$, then it means that vertex $l$ is supported by box $k$. Besides, if a box $i$ relies on a cut, then two vertices are supported. If we want to require four supported vertices, then the factor 3 on the right-hand side of constraints (26) has to be equal to 4.

To define these new variables, constraints (27)–(48) are added.

$$z_{i}\leq(1-g_{i})H,\qquad\forall i, \tag{27}$$

$$z_{k}'-z_{i}\leq v_{ik},\qquad\forall i,k, \tag{28}$$

$$z_{i}-z_{k}'\leq v_{ik},\qquad\forall i,k, \tag{29}$$

$$v_{ik}\leq z_{k}'-z_{i}+2H(1-m_{ik}),\qquad\forall i,k, \tag{30}$$

$$v_{ik}\leq z_{i}-z_{k}'+2Hm_{ik},\qquad\forall i,k, \tag{31}$$

$$h_{ik}\leq v_{ik},\qquad\forall i,k, \tag{32}$$

$$v_{ik}\leq h_{ik}H,\qquad\forall i,k, \tag{33}$$

$$o_{ik}\leq x_{ik}^{p}+x_{ki}^{p}+y_{ik}^{p}+y_{ki}^{p}\leq 2o_{ik},\qquad\forall i,k, \tag{34}$$

$$(1-s_{ik})\leq h_{ik}+o_{ik}\leq 2(1-s_{ik}),\qquad\forall i,k, \tag{35}$$

$$p_{ij}-p_{kj}\leq 1-s_{ik},\qquad\forall i,j,k, \tag{36}$$

$$p_{kj}-p_{ij}\leq 1-s_{ik},\qquad\forall i,j,k, \tag{37}$$

$$\beta_{ik}^{l}\leq s_{ik},\qquad\forall i,k,l, \tag{38}$$

$$\eta_{ik}^{1}+\eta_{ik}^{2}\leq 2(1-\beta_{ik}^{1}),\qquad\forall i,k, \tag{39}$$

$$\eta_{ik}^{2}+\eta_{ik}^{3}\leq 2(1-\beta_{ik}^{2}),\qquad\forall i,k, \tag{40}$$

$$\eta_{ik}^{3}+\eta_{ik}^{4}\leq 2(1-\beta_{ik}^{3}),\qquad\forall i,k, \tag{41}$$

$$\eta_{ik}^{1}+\eta_{ik}^{4}\leq 2(1-\beta_{ik}^{4}),\qquad\forall i,k, \tag{42}$$

$$x_{k}\leq x_{i}+\eta_{ik}^{1}L,\qquad\forall i,k, \tag{43}$$

$$y_{k}\leq y_{i}+\eta_{ik}^{2}W,\qquad\forall i,k, \tag{44}$$

$$x_{i}'\leq x_{k}'+\eta_{ik}^{3}L,\qquad\forall i,k, \tag{45}$$

$$y_{i}'\leq y_{k}'+\eta_{ik}^{4}W,\qquad\forall i,k, \tag{46}$$

$$(1-\gamma_{i}^{1})M\geq z_{i}+a_{1j}x_{i}-b_{1j}-(1-p_{ij})M,\qquad\forall i,j, \tag{47}$$

$$(1-\gamma_{i}^{2})M\geq z_{i}-a_{2j}x_{i}'+b_{2j}-(1-p_{ij})M,\qquad\forall i,j, \tag{48}$$

$i,k\in\{1,\ldots,n\}$, $j\in\{1,\ldots,m\}$, $l\in\{1,\ldots,4\}$.

By constraints (27), if $g_i$ equals 1, then box $i$ is on the ground. Constraints (28)–(33) define the variables $h_{ik}$ by using $v_{ik}$, which represent the absolute value $|z'_k - z_i|$ and $m_{ik}$ that is equal to 1, if $z'_k \geq z_i$ and 0 otherwise. Constraints (34) are based on the fact that boxes $i$ and $k$ share a part of their orthogonal projection on the $XY$ plane if $x^p_{ik} + x^p_{ki} + y^p_{ik} + y^p_{ki} = 0$. A full determination of the variables $o_{ik}$ is required in the hereinafter. If the bottom face of box $i$ is supported by the top face of a box $k$, it implies $h_{ik} + o_{ik} = 0$. This is represented by constraints (35). A full determination of the variables $s_{ik}$ is also required. A box $i$ can be supported by a box $k$ only if these two boxes are in the same bin, which is guaranteed by constraints (36)–(37). Constraints (38) certify that a box $k$ support one vertex of the basis of box $i$ only if this one is supported by box $k$, that is, if $s_{ik} = 1$. The idea of constraints (39)–(42) is depicted in Fig. 9. Constraints (43)–(46) are similar to constraints (15) and (17). Constraints (47) and (48) express that a box $i$ is supported by a cut, if this box satisfies the cut at equality. If the ULD $j$ is a parallelepiped that has no cut, then the value of $\gamma_i^1$ and $\gamma_i^2$ should equal 0 for all the boxes $i$ located in ULD $j$. For this reason, the following constraints are added:

$$p_{ij} + \gamma_i^1 \leq a_{1j} + b_{1j} + 1 \qquad \forall i, j, \tag{49}$$

$$p_{ij} + \gamma_i^2 \leq a_{2j} + b_{2j} + 1 \qquad \forall i, j. \tag{50}$$

*Fragility.* As explained above, some boxes can be fragile, that is, they cannot support boxes on their top face. This can be caused by the nature of the contents of these boxes. To express that a box $i$ is fragile, a new set of parameters is introduced:

$$f_i = \begin{cases} 1 & \text{if box } i \text{ is fragile} \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in \{1, \ldots, n\}.$$

Constraints (51) ensure that if a box is fragile, then it does not support any other box on its top face:

$$\sum_i s_{ik} \leq n(1 - f_k) \qquad \forall k \in \{1, \ldots, n\}. \tag{51}$$

Indeed, the term $\sum_i s_{ik}$ represents the number of boxes supported by box $k$.

*Weight distribution.* In this paragraph, the index $i$, $i \in \{1, \ldots, n\}$ denotes the boxes and the index $j$, $j \in \{1, \ldots, m\}$ denotes the ULDs.

As said in the definition of the problem, we would like to ensure some uniformity about weight distribution. More exactly, according to the control and loading manual of Boeing, the CG of the ULDs must lie within a specific area. Horizontally, this area is defined around the geometric center of the ULD basis. Vertically, the CG must lie below a given level.

Physically speaking, one has

(1) A system of particles moves as if the resultant external force were applied to a single particle of mass $M$ (mass of the system) located at its CG.

(2) The $x$ coordinate of the CG of $n$ particles is defined to be

$$x_{CG} = \frac{\sum_i m_i x_i}{\sum_i m_i}, \tag{52}$$

where $x_i$ is the $x$-coordinate of the $i$th particle and $m_i$ its mass.

As a consequence of the first quotation, each box $i$ can be considered as a point located at the coordinate $(\frac{x_i+x_i'}{2}, \frac{y_i+y_i'}{2}, \frac{z_i+z_i'}{2})$ of mass $m_i$, because the weight is uniformly distributed in the boxes. The second quotation states that the CG of the contents of ULD $j$ is located at the coordinate

$$\frac{1}{\sum\limits_{i|p_{ij}=1} m_i}\left(\sum_{i|p_{ij}=1} m_i(\frac{x_i+x_i'}{2}), \sum_{i|p_{ij}=1} m_i(\frac{y_i+y_i'}{2}), \sum_{i|p_{ij}=1} m_i(\frac{z_i+z_i'}{2})\right) := (x_{CG_j}, y_{CG_j}, z_{CG_j}), \tag{53}$$

the sums being applied only to the boxes $i$ inside ULD $j$.

Here is the approach for the $x$-axis. Since $\frac{L_j}{2}$ is the $x$-coordinate of the geometric center of ULD $j$, we want $x_{CG_j}$ to lie in the neighborhood of $\frac{L_j}{2}$. To define the allowable range of the $x_{CG_j}$, a new parameter $\alpha_j^L$ depending on the type of container and type of aircraft is introduced. Then, $x_{CG_j}$ must fall into the interval $[\frac{L_j}{2} - \alpha_j^L, \frac{L_j}{2} + \alpha_j^L]$. To select only the boxes that are in bin $j$, some new real variables are introduced

$$X_{ij} \equiv p_{ij}\left(\frac{x_i + x_i'}{2}\right).$$

According to this definition, they have to satisfy the following constraints:

$$X_{ij} \le Lp_{ij}, \qquad \forall i, j, \tag{54}$$

$$X_{ij} \le \frac{x_i + x_i'}{2}, \qquad \forall i, j, \tag{55}$$

$$X_{ij} \ge \frac{x_i + x_i'}{2} - L(1 - p_{ij}), \qquad \forall i, j, \tag{56}$$

which are linear. To ensure that $x_{CG_j}$ is in the neighborhood of $L_j/2$, constraints (57) are added.

$$\left(\frac{L_j}{2} - \alpha_j^L\right)\left(\sum_i m_i p_{ij}\right) \le \sum_i X_{ij} m_i \le \left(\frac{L_j}{2} + \alpha_j^L\right)\left(\sum_i m_i p_{ij}\right) \forall j. \tag{57}$$

The weight distribution along the $y$-axis can be managed in the same way. A parameter $\alpha_j^W$ is introduced and $y_{CG_j}$ must lie within the interval $[\frac{W_j}{2} - \alpha_j^W, \frac{W_j}{2} + \alpha_j^W]$. Therefore, the real variables $Y_{ij}$ defined as

$$Y_{ij} \equiv p_{ij}\left(\frac{y_i + y_i'}{2}\right)$$

are introduced. The corresponding constraints, similar to (54)–(57), are

$$Y_{ij} \leq W p_{ij}, \forall i, j, \tag{58}$$

$$Y_{ij} \leq \frac{y_i + y_i'}{2}, \forall i, j, \tag{59}$$

$$Y_{ij} \geq \frac{y_i + y_i'}{2} - W(1 - p_{ij}), \forall i, j, \tag{60}$$

$$\left(\frac{W_j}{2} - \alpha_j^W\right)\left(\sum_i m_i p_{ij}\right) \leq \sum_i Y_{ij} m_i \leq \left(\frac{W_j}{2} + \alpha_j^W\right)\left(\sum_i m_i p_{ij}\right), \forall j. \tag{61}$$

About the weight distribution along the $z$-axis, the reasoning is the same except that the CG can lie as low as possible. Then, a parameter $\alpha_j^H$—which is the maximal value of $z_{CG_j}$—is introduced. This means that $z_{CG_j}$ must lie within the interval $[0, \alpha_j^H]$. Therefore, the real variables $Z_{ij}$ defined as

$$Z_{ij} \equiv p_{ij}\left(\frac{z_i + z_i'}{2}\right)$$

are introduced. The corresponding constraints, similar to (54)–(57), are

$$Z_{ij} \leq H p_{ij}, \quad \forall i, j, \tag{62}$$

$$Z_{ij} \leq \frac{z_i + z_i'}{2}, \quad \forall i, j, \tag{63}$$

$$Z_{ij} \geq \frac{z_i + z_i'}{2} - H(1 - p_{ij}), \quad \forall i, j, \tag{64}$$

$$0 \leq \sum_i Z_{ij} m_i \leq \alpha_j^H \left(\sum_i m_i p_{ij}\right), \quad \forall j. \tag{65}$$

The weight distribution along the $z$-axis can become a problem for the containers with a cut of type 1 or 2.

### 3.7. Complexity

Counting all the variables and constraints described above, the number of variables is $O(n^2)$ and the number of constraints is $O(n^2 m)$.

## 4. Computational experiments

### 4.1. Context

We have tested our mathematical model on a set of small instances. The goal is to check the validity of the model and to get some insights that could help us to develop heuristics for bigger cases. As our

Table 2
Data of LD11 and LD6 ULDs

| | Length (inch) | Width (inch) | Height (inch) | Capacity (kg) | Available volume ($ft^3$) | $\alpha^L$ (inch) | $\alpha^W$ (inch) | $\alpha^H$ (inch) |
|---|---|---|---|---|---|---|---|---|
| LD11 | 125 | 60.4 | 64 | 3125 | 262 | 12.5 | 6 | 34 |
| LD6 | 160 | 60.4 | 64 | 3125 | 322 | 12.5 | 6 | 34 |

The dimensions are expressed in inches as it is the most common unit in the field.

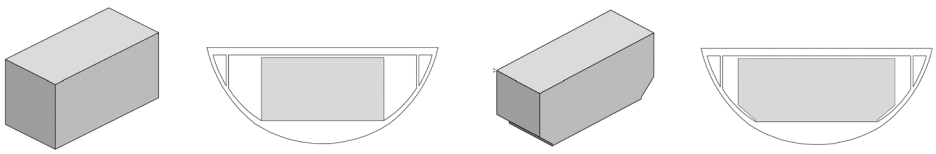

Fig. 10. LD11 ULD on the two pictures on the left-hand side and LD6 ULD on the two pictures on the right-hand side.

approach is based on a mixed integer linear program, it can be handled by a standard optimization library. We rely here on IBM ILOG CPLEX 12.5 with the default parameters. In order to generate the results, we have written a software program in Java. The role of this program is to prepare the data, to call the optimization library, and to analyze the results. It includes a 3D interface that allows us to display the results from different angles and to zoom in and out. The tests were carried out on a personal computer (Windows 8, Intel Core i7, 2.40 GHz, 8.00 GB of RAM). We have limited the computation time to one hour.

We focus on an air transportation application where the containers are ULDs. We specifically work with two very common models: ULDs of type LD11 and LD6. Those two types fit for instance in the lower deck of the Boeing 747 Freighter, one of the most operated cargo aircraft in the world. A full description of these ULDs can be found in the control and loading manual of the Boeing 747-400F (Boeing, 2006). We give their main characteristics in Table 2 and a representation in Fig. 10. Containers of LD11 type are full parallelepipeds, whereas containers of LD6 type are parallelepipeds with a cut of type 1 and type 2.

The mathematical equation describing the cut of type 1 (resp. type 2) is $z = -\frac{166}{175}x + 166$ (resp. $z = \frac{166}{175}x - \frac{236,550}{175}$).

### 4.2. Basic cases

We provide three sets of basic experiments. They differ by the set of available ULDs.

- Series 1: all the ULDs are identical and parallelepipeds.
- Series 2: all the ULDs are identical and not parallelepipeds.
- Series 3: the ULDs can be parallelepipeds or not.

*Series 1*

In order to know beforehand the optimal result and to be able to measure the quality of the solutions, we have generated the box dimensions starting from the definition of the ULDs. We have constructed two sets of instances. In the first one, the whole load can fit exactly inside one LD11 ULD without any unused space. In the second set, we have done the same but with two LD11 ULDs. To get the dimensions of the boxes, the container has been cut, one or several times, along each axis at random locations. Each resulting part is a box. By repeating this procedure with different seed for the random generator and with different numbers of cuts, we have got several data sets. For instance, to get eight boxes of one LD11 ULD, we have drawn at random a value between 0 and $L$ (resp. $W$, $H$) for $l$ (resp. $w$, $h$) and then the dimensions of the boxes are: $l \times w \times h$, $(L-l) \times w \times h$, $l \times (W-w) \times h$, $(L-l) \times (W-w) \times h$, $l \times w \times (H-h)$, $(L-l) \times w \times (H-h)$, $l \times (W-w) \times (H-h)$, $(L-l) \times (W-w) \times (H-h)$. If one tries to optimize the loading of the boxes of the first set inside a pool of containers of type LD11, the optimal solution is clearly to use only one ULD. We have also split one ULD in 12 (two cuts along $X$, one cut along $Y$ and $Z$), 18 (two cuts along $X$ and $Y$, one cut along $Z$), and 27 (two cuts along each axis) boxes. When we have built boxes to fit exactly inside two ULDs, we have considered a mixture of decomposition in eight and 12 boxes. We have generated at random five sets for each of the configurations.

For the sake of simplicity, boxes of series 1 have their weight proportional to their volume, with the same density for all boxes. For instance, it would be approximately the case for boxes filled with a same material. By default, boxes are not fragile and can rotate in all directions. We have also ensured vertical stability with three vertices of the basis supported. We have relaxed these assumptions in Section 4.3.

Although we know that the optimal solution uses only one ULD in the first case and two in the second case, this information has not been used during the optimization. On the contrary, the process has to select the optimal number of ULDs within a pool of three ones. Note that the computation time increases when more ULDs are available. The main results are summarized in Table 3. The first column gives the ID of the instance, the second one the CPU time in seconds, and the third one the number of ULDs selected/required/available. The last column shows the relative CPLEX GAP, that is, the relative difference between the objective value of the best feasible solution and the best-known lower bound. The mathematical model allows us to get feasible results for nearly all these instances. Within the time limit of one hour, the optimal solution is only found for the smallest instances. For the other ones, the third column gives information about the quality of the solution. As all the ULDs have the same volume in this case, the objective function is determined by the number of ULDs selected by the process. One or two additional ULDs are selected in the worst cases.

*Series 2*

We have managed the case of the LD6 truncated containers by decomposing each of them into two parallelepipeds as illustrated in Fig. 11. Then, we have applied separately the same procedure to those parallelepipeds as for the LD11 type. The instances with eight boxes have been composed of both parts cut into four boxes. The instances made of 12 boxes have been built by slicing either the big parallelepiped (in dark grey in Fig. 11) into eight boxes and the small one (in light grey in

Table 3
Results of series 1

| Instance | Time | s/r/a | GAP (%) | Instance | Time | s/r/a | GAP (%) |
|---|---|---|---|---|---|---|---|
| 1ULD × 8 boxes | | | | 2ULDs × 8 boxes | | | |
| 1 | 1.017 | 1/1/3 | 0 | 1 | 38.408 | 2/2/3 | 0 |
| 2 | 0.922 | 1/1/3 | 0 | 2 | 60.509 | 2/2/3 | 0 |
| 3 | 0.532 | 1/1/3 | 0 | 3 | 713.373 | 2/2/3 | 0 |
| 4 | 0.923 | 1/1/3 | 0 | 4 | 317.340 | 2/2/3 | 0 |
| 5 | 1.235 | 1/1/3 | 0 | 5 | 312.452 | 2/2/3 | 0 |
| 1ULD × 12 boxes | | | | (1ULD × 8) + (1ULD × 12) boxes | | | |
| 1 | 26.230 | 1/1/3 | 0 | 1 | 3600.000 | 3/2/3 | 33 |
| 2 | 30.915 | 1/1/3 | 0 | 2 | 1453.679 | 2/2/3 | 0 |
| 3 | 71.357 | 1/1/3 | 0 | 3 | 3600.000 | -/2/3 | – |
| 4 | 5.532 | 1/1/3 | 0 | 4 | 3600.000 | 3/2/3 | 33 |
| 5 | 22.244 | 1/1/3 | 0 | 5 | 3600.000 | 3/2/3 | 33 |
| 1ULD × 18 boxes | | | | 2ULDs × 12 boxes | | | |
| 1 | 559.588 | 1/1/3 | 0 | 1 | 3600.000 | 3/2/3 | 33 |
| 2 | 663.751 | 1/1/3 | 0 | 2 | 3600.000 | -/2/3 | – |
| 3 | 3600.000 | 2/1/3 | 50 | 3 | 3600.000 | 3/2/3 | 33 |
| 4 | 3600.000 | 2/1/3 | 50 | 4 | 3600.000 | -/2/3 | – |
| 5 | 3600.000 | 2/1/3 | 50 | 5 | 3600.000 | -/2/3 | – |
| 1ULD × 27 boxes | | | | | | | |
| 1 | 3600.000 | 2/1/3 | 50 | | | | |
| 2 | 3600.000 | –/1/3 | – | | | | |
| 3 | 3600.000 | –/1/3 | – | | | | |
| 4 | 3600.000 | –/1/3 | – | | | | |
| 5 | 3600.000 | –/1/3 | – | | | | |

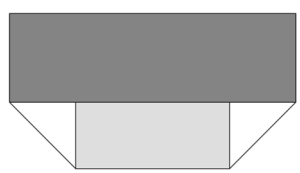

Fig. 11. Decomposition of LD6.

Fig. 11) in four boxes, or the opposite. The former five instances are denoted "4s + 8B" in Table 4 and the latter instances are denoted "8s + 4B." For the last case, both parallelepipeds have been cut into eight boxes.

The main results are summarized in Table 4. The first column gives the ID of the instance, the second one the CPU time in seconds, and the third one the number of ULDs selected/required/available. The last column shows the CPLEX GAP. All the solutions satisfy the constraints. However, this series is more complex and the process is not always able to find a feasible solution within the time limit. The small instances remain manageable.

Table 4
Results of series 2

| Instance | Time | s/r/a | GAP (%) | Instance | Time | s/r/a | GAP (%) |
|---|---|---|---|---|---|---|---|
| 1ULD × (4 + 4) boxes | | | | 1ULD × (4s + 8B) boxes | | | |
| 1 | 1.952 | 1/1/3 | 0 | 1 | 3600.000 | –/1/3 | – |
| 2 | 1.534 | 1/1/3 | 0 | 2 | 3600.000 | –/1/3 | – |
| 3 | 3.113 | 1/1/3 | 0 | | 1278.400 | 1/1/3 | 0 |
| 4 | 3.352 | 1/1/3 | 0 | 4 | 133.880 | 1/1/3 | 0 |
| 5 | 1.994 | 1/1/3 | 0 | 5 | 2736.852 | 1/1/3 | 0 |
| 1ULD × (8s + 4B) boxes | | | | 1ULD × (8 + 8) boxes | | | |
| 1 | 3600.000 | –/1/3 | – | 1 | 3600.000 | –/1/3 | – |
| 2 | 44.885 | 1/1/3 | 0 | 2 | 3600.000 | –/1/3 | – |
| 3 | 235.490 | 1/1/3 | 0 | 3 | 3600.000 | –/1/3 | – |
| 4 | 101.724 | 1/1/3 | 0 | 4 | 3600.000 | –/1/3 | – |
| 5 | 3600.000 | –/1/3 | – | 5 | 3600.000 | –/1/3 | – |

*Series 3*

For the third series, we have generated more complex configurations by mixing the overall dimensions of both types of containers. We have first started by defining a virtual container as the smallest parallelepiped able to contain a LD6 container. Then, we have sliced this virtual container into boxes using the same methodology as the one used for the LD11 type. We have built instances with 8, 12, and 18 boxes. This implies that two containers are required in the optimal solution. However, in this case neither their type nor the unused volume can be predicted. In order to strengthen the problem, we have completed these sets with one additional box generated randomly so that it can fit inside an LD11 container.

For each instance, the optimization process worked with a pool of two LD6 and two LD11 ULDs. The main results are summarized in Table 5. The first column gives the ID of the instance; the second one the CPU time in seconds; the third one the number, type, and filling rate of the ULDs selected in the solution; and the fourth column represents the CPLEX GAP. Considering both types of containers increases the computation times. Figure 12 shows the configuration of the solution found for the second instance of 13 boxes. These pictures are screenshots of the 3D interface.

### *4.3. Additional cases*

We present some variants of the previous cases. We relax several specific constraints one at a time. The detailed results are available in the Appendix.

*Fragility*

To test the impact of the fragility constraints, we have considered some new sets of five instances composed of 12 boxes randomly cut out of an LD11 ULD. Each box has a given probability (15%, 20%, 25%, 30%, 40%, 50%) to be fragile. The number of fragile boxes seems to increase slightly the computation times for some instances and sometimes the optimal solution cannot be reached within an hour. It also seems to influence the number of selected ULDs. Indeed, when the

Table 5
Results of series 3

| Instance | Time | Filling rate | GAP (%) |
|---|---|---|---|
| **9 boxes** | | | |
| 1 | 18.839 | 2 LD11s (68.56%; 59.57%) | 0 |
| 2 | 3.033 | 2 LD11s (68.23%; 60.15%) | 0 |
| 3 | 30.824 | 2 LD11s (51.91%; 76.32%) | 0 |
| 4 | 4.018 | 2 LD11s (69.03%; 60.24%) | 0 |
| 5 | 3600.000 | 1 LD6 (42.42%); 2 LD11s (39.84%; 37.42%) | 36.87 |
| **13 boxes** | | | |
| 1 | 29.960 | 2 LD11s (62.04%; 66.18%) | 0 |
| 2 | 35.768 | 2 LD11s (36.57%; 97.85%) | 0 |
| 3 | 157.055 | 2 LD11s (85.45%; 51.91%) | 0 |
| 4 | 11.712 | 2 LD11s (58.96%; 86.77%) | 0 |
| 5 | 67.867 | 2 LD11s (733.84%; 63.31%) | 0 |
| **19 boxes** | | | |
| 1 | 3600.000 | – | – |
| 2 | 3600.000 | – | – |
| 3 | 3600.000 | – | – |
| 4 | 3600.000 | – | – |
| 5 | 3600.000 | – | – |

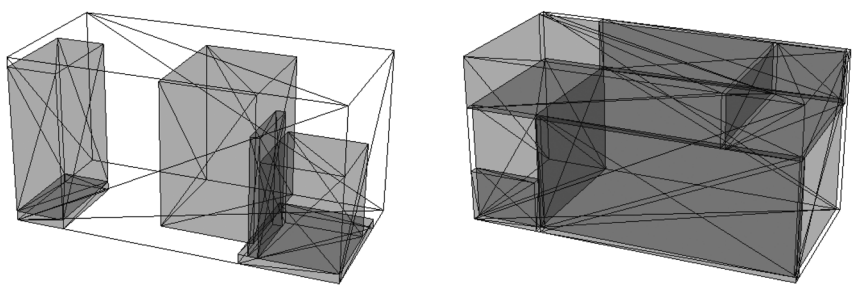

Fig. 12. Optimal solution of the second instance of 13 boxes of the series 3.

number of fragile boxes is large, there is sometimes no more possibility to stack them and we have to use more ULDs.

*Orientation*

Some boxes may not be allowed to rotate in some directions. To express this situation, parameters $l_i^+$, $w_i^+$, and $h_i^+$ have been introduced in the formulation. We have built three new instances composed of 12 nonfragile boxes randomly cut out of an LD11 ULD. For each box of those instances, each of the three parameters is set to zero, that is, the corresponding side cannot be in a vertical position, according to a predefined probability (10%, 20% or 30%). For few instances, the optimal solution cannot be reached within an hour. As for the fragility, the number of forbidden rotations seems to influence the number of selected ULDs.

*Weight distribution*

We initially make the assumption that the weight of a box is proportional to its volume and that the density is the same for each box. The weight distribution constraint is easier to satisfy in this case since the whole load is uniform. Let us now imagine that all boxes can have different types of content and different densities, leading to a more heterogeneous assortment. In the new instances, the 12 nonfragile boxes, which have randomly cut out of a LD11 ULD, can be made of material with three distinct densities. For each box, the density has been drawn at random. Two triplets of possible densities have been considered. The difference between the three densities of the first set is smaller than the one between the three densities of the second set.

For each instance of the sets, three LD11 ULDs are available. The optimization process, based on our model, always found the optimal solutions. For these instances, only one ULD is necessary to pack all the boxes and the computation times do not seem to be influenced by the different densities of the load.

*High vertical stability*

In Section "Specific constraints," the stability is achieved by supporting either three or four vertices of each box. Three vertices are sufficient most of the time. It was the case with all the previous experiments. Requesting four vertices is stronger and more restrictive. In this paragraph, we measure the impact of this choice. The configuration of the new experiments is the same as the one for series 1 and 3, accepted for the number of supported vertices requested. Let us note that the optimal solutions of the first series satisfy more easily the stability constraints with four vertices than the ones of the third series.

In comparison with the results of the series 1, the instances with eight or 12 boxes are solved within the same computation times. For the sets of instances with 18 and 27 boxes, some instances are solved while they were not in the series 1. It seems to be the same for the series 3. More tests should be carried out to confirm whether this is the tendency or not .

No major conclusion can be drawn based on these small tests. Indeed, the aim is to show the diversity and flexibility of the developed model. However, these observations, and some insights on the branching scheme, may help to define heuristics in a future work. Note that all the instances can be found on the following website (http://www.quantom.hec.ulg.ac.be/projects.php).

## 5. Conclusions

The bin packing problem is a current problem encountered in transport. This paper presents a mixed integer programming formulation for the 3D bin packing problem deriving from an air cargo application. This formulation deals with some constraints inherent to transport such as cargo stability and fragility as well as some conditions on the weight distribution. Based on an air cargo application, another distinctive feature is the special shape of the containers, which are called the ULDs. Some of these shapes can influence the stability of the cargo. Our contribution consists in developing a new mathematical model taking these specificities into account. Besides minimizing the unused space inside the containers, it also proposes an actual loading pattern. We have tested this formulation on small instances and looked at some variants of the problem.

The MBSBPP problem is known to be NP-hard, opening the way to heuristics. The obtained results would help to test the more suitable techniques and new procedures combining heuristics and exact algorithms.

## Acknowledgments

The project that leads to these results was partially funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (grant P7/36). This paper, however, only expresses the authors' views. The authors would like to thank José Fernando Oliveira for his valuable advice. They would also like to thank the two anonymous reviewers whose comments and suggestions helped to improve and clarify this article.

## References

Almeida, A.D., Figueiredo, M.B., 2010. A particular approach for the three-dimensional packing problem with additional constraints. *Computers & Operations Research* 37, 11, 1968–1976.

Amiouny, S.V., Bartholdi, J.J., Vande Vate, J.H., Zhang, J., 1992. Balanced loading. *Operations Research* 40, 2, 238–246.

Baldi, M.M., Perboli, G., Tadei, R., 2012. The three-dimensional knapsack problem with balancing constraints. *Applied Mathematics and Computation* 218, 19, 9802–9818.

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

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

Boeing, 2006. Weight and balance control and loading manual—sample manual, model 747-430.

Bortfeldt, A., Wäscher, G., 2013. Constraints in container loading—a state-of-the-art review. *European Journal of Operational Research* 229, 1–20.

Brunetta, L., Gregoire, P., 2005. A general purpose algorithm for three-dimensional packing. *INFORMS Journal on Computing* 17, 3, 328–338.

Ceschia, S., Schaerf, A., 2013. Local search for a multi-drop multi-container loading problem. *Journal of Heuristics* 19, 2, 275–294.

Chan, F.T.S., Bhagwat, R., Kumar, N., Tiwari, M., Lam, P., 2006. Development of a decision support system for air-cargo pallets loading problem: a case study. *Expert Systems with Applications* 31, 472–485.

Chen, C., Lee, S., Shen, Q., 1995. An analytical model for the container loading problem. *European Journal of Operational Research* 80, 68–76.

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

Davies, A., Bischoff, E., 1999. Weight distribution considerations in container loading. *European Journal of Operational Research* 114, 209–527.

Fok, K., Chun, A., 2004. Optimizing air cargo load planning and analysis. *Proceedings of the International Conference on Computing, Communications and Control Technologies*, Austin, Texas.

Gehring, H., Bortfeldt, A., 1997. A genetic algorithm for solving the container loading problem. *International Transactions in Operational Research* 4, 5–6, 401–418.

Hadjiconstantinou, E., Christofides, N., 1995. An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of Operational Research* 83, 1, 39–56.

Herz, J., 1972. Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development* 16, 462–469.

Jin, Z., Ito, T., Ohno, K., 2003. A three-dimensional bin packing problem and its practical algorithm. *JSME International Journal Series C: Mechanical Systems, Machine Elements and Manufacturing* 46, 1, 60–66.

Junqueira, L., Morabito, R., Yamashita, D.S., 2012. Three-dimensional container loading models with cargo stability and load bearing constraints. *Computers and Operations Research* 39, 74–85.

Kaluzny, B.L., Shaw, R.H.A.D., 2009. Optimal aircraft load balancing. *International Transactions in Operational Research* 16, 6, 767–787.

Limbourg, S., Schyns, M., Laporte, G., 2012. Automatic aircraft cargo load planning. *Journal of the Operational Research Society* 63, 1271–1283.

Lin, J.-L., Chang, C.-H., Yang, J.-Y., 2006. A study of optimal system for multiple-constraint multiple-container packing problems. In Ali, M., Dapoigny, R. (eds) *Advances in Applied Artificial Intelligence, Vol. 4031 of Lecture Notes in Computer Science*. Springer, Berlin-Heidelberg, pp. 1200–1210.

Martello, S., Pisinger, D., Vigo, D., 2000. The three-dimensional bin packing problem. *Operations Research* 48, 2, 256–267.

Mongeau, M., Bès, C., 2003. Optimization of aircraft container loading. *IEEE Transactions on Aerospace and Electronic Systems* 39, 140–150.

Moon, I., Nguyen, T., 2013. Container packing problem with balance constraints. *OR Spectrum*, doi: 10.1007/s00291-013-0356-1.

Padberg, M., 2000. Packing small boxes into a big box. *Mathematical Methods of Operations Research* 52, 1–21.

Techanitisawad, A., Tangwiwatwong, P., 2004. A GA-based heuristic for the interrelated container selection loading problems. *Industrial Engineering and Management Systems* 3, 22–37.

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

Tsai, R., Malstrom, E., Kuo, W., 1993. Three dimensional palletization of mixed box sizes. *IIE Transactions* 25, 4, 64–75.

Vancroonenburg, W., Verstichel, J., Tavernier, K., Vanden Berghe, G., 2014. Automatic air cargo selection and weight balancing: a mixed integer programming approach. *Transportation Research E, Logistics and Transportation Review* 65, 70–83.

Wäscher, G., Haußner, H., Schumann, H., 2007. An improved typology of cutting and packing problems. *European Journal of Operational Research* 183, 1109–1130.

Westerlund, J., Papageorgiou, L., Westerlund, T., 2005. *A Problem Formulation for Optimal Mixed-Sized Box Packing, Vol. 20 of Computer Aided Chemical Engineering*, Elsevier, Barcelona.

## Appendix

### *Fragility*

Table A.1 contains the results of experiments with fragile boxes. The first column represents the ID of the instance, the second one the actual number of fragile boxes, the third one the CPU time in seconds, the fourth one the number of selected LD11s, the fifth column the CPLEX GAP, and the sixth one the filling rate of the selected ULDs.

### *Orientation*

Table A.2 contains the results of experiments for which some boxes are not allowed to rotate along the vertical axis. The first column represents the ID of the instance, the second one the CPU time in seconds, the third one the number of selected LD11s, the fourth column the CPLEX GAP, and the fifth one the filling rate of the selected ULDs.

### *Weight distribution*

The density of each box is drawn at random. Table A.3 shows the impact on the results. The first column represents the ID of the instance, the second one the CPU time in seconds, the third one

Table A.1
Numerical results for fragile boxes

| | Fragility | Time | Selected | GAP (%) | Filling (%) | | Fragility | Time | Selected | GAP (%) | Filling (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Probability of fragility: 15%** | | | | | | **Probability of fragility: 20%** | | | | | |
| 1 | 2 | 27.339 | 1 | 0 | 100 | 1 | 2 | 14.891 | 1 | 0 | 100 |
| 2 | 1 | 986.307 | 1 | 0 | 100 | 2 | 3 | 3.551 | 2 | 0 | 100 |
| 3 | 2 | 8.928 | 1 | 0 | 100 | 3 | 2 | 35.476 | 2 | 0 | 80.99; 19.01 |
| 4 | 1 | 25.592 | 1 | 0 | 100 | 4 | 3 | 29.172 | 1 | 0 | 100 |
| 5 | 2 | 60.93 | 1 | 0 | 100 | 5 | 3 | 32.155 | 1 | 0 | 100 |
| **Probability of fragility: 25%** | | | | | | **Probability of fragility: 30%** | | | | | |
| 1 | 3 | 3600.000 | 2 | 50 | 61.88; 38.12 | 1 | 4 | 33.164 | 2 | 0 | 81.69; 18.31 |
| 2 | 3 | 33.018 | 1 | 0 | 100 | 2 | 3 | 1981.831 | 2 | 0 | 77.58; 22.42 |
| 3 | 3 | 25.970 | 1 | 0 | 100 | 3 | 3 | 19.540 | 1 | 0 | 100 |
| 4 | 3 | 3600.000 | 2 | 50 | 60.85; 39.15 | 4 | 3 | 28.297 | 1 | 0 | 100 |
| 5 | 3 | 29.737 | 1 | 0 | 100 | 5 | 4 | 1328.851 | 2 | 0 | 25.41 74.59 |
| **Probability of fragility: 40%** | | | | | | **Probability of fragility: 50%** | | | | | |
| 1 | 4 | 8.870 | 1 | 0 | 100 | 1 | 6 | 23.201 | 2 | 0 | 75.51; 24.49 |
| 2 | 4 | 10.600 | 1 | 0 | 100 | 2 | 6 | 594.620 | 2 | 0 | 39.72; 60.28 |
| 3 | 5 | 79.486 | 2 | 0 | 58.02; 41.98 | 3 | 6 | 136.131 | 2 | 0 | 27.61; 72.39 |
| 4 | 5 | 70.151 | 2 | 0 | 77.87; 22.13 | 4 | 6 | 58.893 | 2 | 0 | 61.17; 38.83 |
| 5 | 5 | 30.700 | 2 | 0 | 13.99; 86.01 | 5 | 6 | 12.996 | 2 | 0 | 23.90; 76.10 |

Table A.2
Numerical results for forbidden rotations

| | Time | Selected | GAP (%) | Filling (%) | | Time | Selected | GAP (%) | Filling (%) |
|---|---|---|---|---|---|---|---|---|---|
| **Probability of nonverticality: 10%** | | | | | **Probability of nonverticality: 20%** | | | | |
| 1 | 16.065 | 2 | 0 | 66.43; 33.57 | 1 | 3600.000 | 2 | 50 | 13.50; 86.50 |
| 2 | 13.929 | 2 | 0 | 81.94; 18.06 | 2 | 119.821 | 2 | 0 | 16.97; 83.03 |
| 3 | 36.183 | 2 | 0 | 33.27; 66.73 | 3 | 123.478 | 2 | 0 | 17.63; 82.37 |
| 4 | 96.916 | 2 | 0 | 64.54; 35.46 | 4 | 271.901 | 2 | 0 | 59.82; 40.18 |
| 5 | 44.721 | 2 | 0 | 47.55; 52.45 | 5 | 57.507 | 2 | 0 | 51.39; 48.61 |
| **Probability of nonverticality: 30%** | | | | | | | | | |
| 1 | 4.113 | 2 | 0 | 58.21; 41.79 | | | | | |
| 2 | 3600.000 | 2 | 50 | 49.59; 50.41 | | | | | |
| 3 | 3.715 | 2 | 0 | 40.45; 59.55 | | | | | |
| 4 | 27.622 | 2 | 0 | 50.63; 49.37 | | | | | |
| 5 | 70.36 | 2 | 0 | 27.56; 72.44 | | | | | |

the number of LD11 selected, the fourth column the CPLEX GAP, and the filling rate in last column.

*High vertical stability*

Four vertices are required to support each box. This is compared with the first and third series for which only three vertices were required to be supported. The first column of Table A.4 gives the ID of the instance, the second one the CPU time in seconds, the third one the number of selected ULDs, and the fourth column represents the CPLEX GAP.

Table A.3
Numerical results for different densities

| | Time | Selected | GAP (%) | Filling (%) |
|---|---|---|---|---|
| $density_0$=150,000 | $density_1$=200,000 | $density_2$=250,000 | | |
| 1 | 36.737 | 1 | 0 | 100 |
| 2 | 41.848 | 1 | 0 | 100 |
| 3 | 35.491 | 1 | 0 | 100 |
| 4 | 30.742 | 1 | 0 | 100 |
| 5 | 19.814 | 1 | 0 | 100 |
| $density_0$=150,000 | $density_1$=300,000 | $density_2$=450,000 | | |
| 1 | 46.151 | 1 | 0 | 100 |
| 2 | 37.187 | 1 | 0 | 100 |
| 3 | 407.100 | 1 | 0 | 100 |
| 4 | 180.541 | 1 | 0 | 100 |
| 5 | 158.599 | 1 | 0 | 100 |

Table A.4
Numerical results for four supported vertices

| Instance | Time | Selected | GAP (%) | Instance | Time | Selected | GAP (%) |
|---|---|---|---|---|---|---|---|
| **Series 1** | | | | **Series 3** | | | |
| 8 boxes | | | | 9 boxes | | | |
| 1 | 0.812 | 1 | 0 | 1 | 6.623 | 2 LD11s (59.44%; 68.70%) | 0 |
| 2 | 0.814 | 1 | 0 | 2 | 7.850 | 2 LD11s (52.25%; 76.12%) | 0 |
| 3 | 0.923 | 1 | 0 | 3 | 26.152 | 2 LD11s (53.28%; 74.95%) | 0 |
| 4 | 1.815 | 1 | 0 | 4 | 2.846 | 2 LD11s (69.03%; 60.24%) | 0 |
| 5 | 0.830 | 1 | 0 | 5 | 3600.000 | 2 LD11s (51.27%; 14.66%); 1 LD6 (51.91%) | 30.96 |
| 12 boxes | | | | 13 boxes | | | |
| 1 | 29.419 | 1 | 0 | 1 | 2381.134 | 2 LD11s (44.19%; 84.97%) | 0 |
| 2 | 6.824 | 1 | 0 | 2 | 125.496 | 2 LD11s (58.37%; 76.06%) | 0 |
| 3 | 50.581 | 1 | 0 | 3 | 31.999 | 2 LD11s (64.38%; 72.98%) | 0 |
| 4 | 3.191 | 1 | 0 | 4 | 43.42 | 2 LD11s (62.44%; 83.29%) | 0 |
| 5 | 27.395 | 1 | 0 | 5 | 18.459 | 2 LD11s (79.74%; 57.42%) | 0 |
| 18 boxes | | | | 19 boxes | | | |
| 1 | 516.052 | 1 | 0 | 1 | 3600.000 | 2 LD11s (37.17%; 42.09%); 1 LD6 (40.61%) | 69.03 |
| 2 | 354.786 | 1 | 0 | 2 | 1754.217 | 2 LD11s (81.36%; 58.76%) | 0 |
| 3 | 3600.000 | 2 | 50 | 3 | 3600.000 | – | – |
| 4 | 3600.000 | 2 | 50 | 4 | 3600.000 | – | – |
| 5 | 1050.527 | 1 | 0 | 5 | 3600.000 | – | – |
| 27 boxes | | | | | | | |
| 1 | 3600.000 | 2 | 50 | | | | |
| 2 | 3600.000 | – | – | | | | |
| 3 | 3600.000 | – | – | | | | |
| 4 | 3600.000 | 3 | 66 | | | | |
| 5 | 3600.000 | – | – | | | | |