MIP-based constructive heuristics for the three-dimensional Bin Packing Problem with transportation constraints

Célia Paquay, Sabine Limbourg, Michaël Schyns & José Fernando Oliveira

Published online: 28 Jul 2017.

Submit your article to this journal ⏎

View related articles ⏎

View Crossmark data ⏎

MIP-based constructive heuristics for the three-dimensional Bin Packing Problem with transportation constraints

Célia Paquay[a]*, Sabine Limbourg[a], Michaël Schyns[a] and José Fernando Oliveira[b]

[a]*HEC Management School, QuantOM, University of Liege (ULg), Liege, Belgium;* [b]*Faculdade de Engenharia, INESC TEC, Universidade do Porto, Porto, Portugal*

This article is about seeking a good feasible solution in a reasonable amount of computation time to the three-dimensional Multiple Bin Size Bin Packing Problem (MBSBPP). The MBSBPP studied considers additional constraints encountered in real world air transportation situations, such as cargo stability and the particular shape of containers. This MBSBPP has already been formulated as a Mixed Integer linear Programming problem, but as yet only poor results have been achieved for even fairly small problem sizes. The goal of the work this paper describes is to develop heuristics that are able to quickly provide good initial feasible solutions for the MBSBPP. Three methodologies are considered, which are based on the decomposition of the original problem into easier subproblems: the matheuristics Relax-and-Fix, Insert-and-Fix and Fractional Relax-and-Fix. They have been parametrised on real data sets and then compared to each other. In particular, two of these techniques show promising results in reasonable computational times.

Keywords: packing; matheuristics; relax-and-fix; transportation constraints; Unit Load Device

1. Introduction

This paper concerns the selection of bins in order to pack a set of cuboid boxes. The aim is to minimise the unused space inside the selected bins. The set of boxes is highly heterogeneous, while there are few types of bins to select. According to the typology of cutting and packing problems developed by Wäscher, Haußner, and Schumann (2007), this is a three-dimensional Multiple Bin Size Bin Packing Problem (MBSBPP).

In addition to geometry constraints which require that boxes lie entirely inside bins and without overlapping other boxes, this paper also considers additional constraints encountered in practical packing situations. These situations include the bin weight limit, orientation constraints, load stability, fragility of the boxes, and weight distribution within a bin. Moreover, as the original problem is an air cargo application, we extend the definition of the MBSBPP to include situations in which the bins may be truncated parallelepipeds. Indeed, in this context, bins are called Unit Load Devices (ULD). A ULD is an assembly of components consisting of a container, or pallet covered with a net, which provides standardised size units for individual pieces of baggage or cargo, and allows for rapid loading and unloading (Limbourg, Schyns, and Laporte 2012). ULDs may have specific shapes to fit inside aircraft, as shown in Figure 1. More details about these constraints can be found in Paquay, Schyns, and Limbourg (2016).

The objective of this paper is to develop a set of constructive matheuristics that are based on the formulation provided by Paquay, Schyns, and Limbourg (2016), which is the first article to provide a Mixed Integer linear Programming problem (MIP) with the constraints mentioned above. This MIP has been tested with a standard Branch-and-Bound (B&B) technique on generated instances and, as expected, the computational times were rather big. For this reason, the goal of this work is to develop heuristics that may be able to provide good quality solutions in short computational times. Our main contribution is to adapt the Relax-and-Fix (R&F) heuristic presented in Pochet and Wolsey (2006) to the MIP from Paquay, Schyns, and Limbourg (2016). The R&F methodology is an iterative procedure, which decomposes a large scale MIP problem into several easier subproblems in order to quickly get an initial feasible solution for the original problem, or to compute bounds on the optimal value. The integrality restriction of some variables is relaxed in the subproblems, reducing the computational times. To our knowledge, this algorithm was originally created to solve Lot-Sizing Problems and has never been applied to Bin Packing Problems. The R&F approach can be regarded as a solution framework and its use to solve a different problem requires extensive research, as it is usual in general solution frameworks. Moreover, considering the way how this solution methodology works, the application to our specific three-dimensional MBSBPP was not straightforward and

---

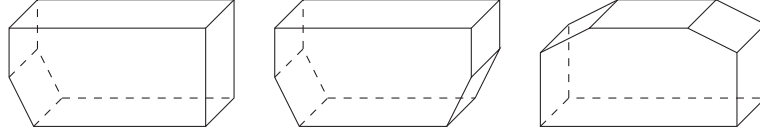*Corresponding author. Email: cpaquay@ulg.ac.be

Figure 1.   Different shapes of Unit Load Devices, the typical containers in air transportation.

required significant modifications. Thus, one result of this work was to adapt the traditional R&F, to quickly find a feasible solution to the problem described above.

In the present paper, two other MIP-based constructive heuristics, the Insert-and-Fix (I&F) and Fractional Relax-and-Fix (FRF), inspired by the R&F heuristic, have also been extended to our context. They were introduced in Liberalino (2012) to solve Lot-Sizing and Vehicle Routing Problems. To our knowledge, these two matheuristics are not common in the literature and thus their application constitutes a definite novelty. As for the R&F methodology, our contribution is to modify these heuristics in order to quickly obtain a heuristic solution to this specific MBSBPP. The I&F algorithm also iteratively builds an initial solution by decomposing the original problem into smaller ones. In this methodology, subproblems are easier to handle because they do not consider all the variables and constraints of the original problem, but the integer constraint is not relaxed. In terms of Lot-Sizing, the variables and constraints inherent to several future time periods are ignored during the first steps of the algorithm. Finally, the FRF heuristic, also called *time decomposition* or *time partitioning* in Pochet and Wolsey (2006), can be seen as an amalgamation of the two previous methods: not all the variables are considered during the first steps (as in the I&F), and some of these variables have the integrality restriction relaxed (as in the R&F).

On the one hand, although the R&F, I&F, and FRF heuristics have never been applied to the three-dimensional MBSBPP, many works have been published about other approaches over recent decades. Among others, Dowsland (1991) improved techniques applied in two-dimensional packing problems to three-dimensional situations. Later, Chen, Lee, and Shen (1995) developed a mathematical formulation for the same problem, but considered only orientation and weight distribution constraints. Faroe, Pisinger, and Zachariasen (2003) developed a guided local search for the three-dimensional MBSBPP and used it to improve the upper bounds. Chan et al. (2006) presented a two-phase intelligent Decision Support System for the three-dimensional MBSBPP, also in the air cargo context, but did not consider the possible fragility of boxes. Lin, Chang, and Yang (2006) combined clustering methods and a genetic algorithm to solve the three-dimensional MBSBPP with orientation, stability, and fragility constraints. Ceschia and Schaerf (2013) considered the three-dimensional MBSBPP arising from an industrial application. They developed local search metaheuristics considering the maximal capacity of the bins, and orientation, fragility, and stability constraints. Other papers also proposed solution techniques for similar problems, such as the Cutting Stock Problem in which there are only a few types of boxes (e.g. Brunetta and Gregoire 2005; Che et al. 2011), or also the Single Knapsack Problem in which there is only one bin to fill with a selection of items (e.g. Ngoi, Tay, and Chua 1994; Hemminki, Leipala, and Nevalainen 1998). More details and literature on the MBSBPP are provided in Bortfeldt and Wäscher (2013).

On the other hand, the R&F matheuristic has already been successfully used in different areas. Among others, Kelly and Mann (2004) apply the R&F procedure to develop a flowsheet decomposition heuristic. Beraldi et al. (2008) deal with an identical parallel machine Lot-Sizing and Scheduling Problem with sequence-dependent set-up costs. Oliveira et al. (2014) propose an R&F heuristic procedure to solve a specific assignment model for vehicle reservation. Baena, Castro, and González (2015) adapt the R&F methodology to the Controlled Tabular Adjustment (CTA), which is a technique for tabular data protection.

This paper is organised as follows: Section 2 provides a summary of the variables of the formulation developed in Paquay, Schyns, and Limbourg (2016). Section 3 describes the three MIP-based constructive heuristics. Section 4 contains the computational results. Finally, conclusions are drawn in Section 5.

2. Mathematical formulation

The constructive heuristics developed in this paper are based on the formulation from Paquay, Schyns, and Limbourg (2016). Thus, the variables of that formulation are of crucial importance to the adaptation process, and for this reason a concise description of them is given below.

The problem is as follows: the objective is to minimise the unused space inside the selected ULDs, considering that (1) each box is assigned to exactly one used ULD, (2) the total weight of the boxes inside each ULD respects its maximum weight capacity, (3) each box lies within the limits of a ULD even when it has a special shape, (4) boxes can be orthogonally
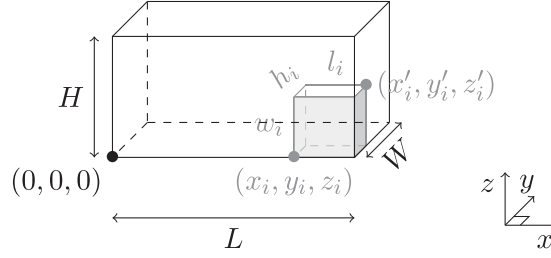
Figure 2. The coordinate system associated to a ULD and the coordinates of a box i.

rotated but some orientations are not permitted for some boxes, (5) there is no overlap of boxes, (6) the packing is stable, (7) fragile boxes cannot support any other boxes, and (8) the weight inside the loaded ULDs has to be uniformly distributed.

Hereinafter, the index j denotes the ULDs, and indices i and k denote the boxes.

The problem can be stated as follows: a set of n cuboid 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 different available ULDs of dimensions $L_j \times W_j \times H_j$, with a maximal weight capacity C_j, and a volume V_j ($j \in \{1, \ldots, m\}$).

The main goal is to select ULDs such as to minimise the unused volume after packing the set of boxes. This selection is represented by the set of binary variables u_j. It describes whether a ULD j is used. Therefore, the objective function can be written as

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

which is equivalent to minimise $\sum_{j=1}^{m} u_j V_j$, since the second term is a constant.

The assignment of a box i to a ULD j is defined by the set of binary variables p_{ij}.

Additionally, loading patterns are also provided. To this purpose, the position of each box i is given by two triplets of integer variables (x_i, y_i, z_i), (x'_i, y'_i, z'_i) representing the front left bottom corner and rear right top corner of box i. The coordinate system has its origin on the front left bottom corner of the ULDs, the x-axis (resp. y-axis, z-axis) lying on the length (resp. width, height) of the ULD, as shown in Figure 2. These triplets of variables are also useful to guarantee that the boxes lie entirely within the ULDs.

Boxes are allowed to orthogonally rotate inside ULDs. Hence, additional binary variables r_{iab}, where a and b describe the axes of coordinates and the sides of box i respectively, are needed to determine the two sets of coordinates of each box i. In more details, the index b indicates the side of the box, i.e. $b \in \{l := 1, w := 2, h := 3\}$, whereas a indicates the axis, i.e. $a \in \{x := 1, y := 2, z := 3\}$. They specify which side of the box i is along which axis.

In some specific situations, the contents of a box can prevent it from being rotated. To represent this situation, three parameters l_i^+, w_i^+ and h_i^+ are introduced for each box i: l_i^+ (resp. w_i^+, h_i^+) equals 1 if the length (resp. width, height) of box i could be in a vertical position, 0 otherwise. These orientation constraints are also handled by the r_{i3b} variables.

To prevent the overlapping of boxes, the relative position of each pair of boxes is defined by the sets of binary variables x_{ik}^p, y_{ik}^p and z_{ik}^p. Variables x_{ik}^p (resp. y_{ik}^p, z_{ik}^p) describe whether box i is on the right side (resp. behind, above) of box k.

Some ULDs may have a special shape, such as inclined walls as if they were cut on the corner, as shown in Figure 1. For this reason, parameters are introduced to describe the four potential cuts of each ULD. Note that a number is assigned to each corner in the xz-plane: the left bottom (resp. right bottom, right top, left top) corner is a cut of type 1 (resp. 2, 3, 4).

The generated loading pattern has to be vertically stable, that is, boxes should not move with respect to the z-axis in a static situation. In Paquay, Schyns, and Limbourg (2016) and in this work, a box is considered vertically stable if it lies on the ground (represented by binary variables g_i), or if its four base vertices are supported by other boxes (defined by binary variables h_{ik}, m_{ik}, o_{ik}, s_{ik}, η_{ik}^1, η_{ik}^2, η_{ik}^3, η_{ik}^4, β_{ik}^l, and integer variables a_{ik}), or by the inclined wall of some ULDs (described by binary variables γ_i^1, γ_i^2).

Due to the contents of the boxes, it may happen that a box, declared fragile, cannot support other boxes. This is handled through variables s_{ik}.

To describe the position of the centre of gravity of the contents of each ULD, three sets of real variables X_{ij}, Y_{ij}, and Z_{ij} are introduced. They are used to ensure that the centre of gravity of the loaded ULDs lies within a specific area along the xy-plane and does not exceed a maximum height.

For the sake of clarity, the different sets of variables are called *families* hereinafter. For instance, we consider u_j family to stand for all u_j variables for all $j \in \{1, \ldots, m\}$.

Table 1.   Blocks tested in the Relax-and-Fix heuristic.

Families in the block	Name given to the block
p_{ij}	p_{ij}
$p_{ij}, x_i, y_i, z_i, x_i', y_i', z_i'$	Coord & p_{ij}
$x_i, y_i, z_i, x_i', y_i', z_i'$	Coord
$p_{ij}, x_i, y_i, z_i, x_i', y_i', z_i', r_{iab}, g_i, \gamma_i^1, \gamma_i^2$	All i

3. MIP-based constructive heuristics

We decided to explore three constructive heuristics, because the MIP formulation provided in Paquay, Schyns, and Limbourg (2016) includes many integer variables, which partially explains the slowness of its resolution. The chosen heuristics can be used to build up initial solutions while utilising the potential offered by this model. As mentioned already, these matheuristics have never been applied to this type of problem. The contribution of this paper consists in extending them with respect to the structure of the MBSBPP.

3.1 *Relax-and-Fix heuristic*

The first in-depth explanation of the R&F methodology for logistics problems can be found in Pochet and Wolsey (2006). The methodology comprises sequentially solving R smaller MIPs to find a heuristic solution to the original MIP. These subproblems, denoted MIP^r for $1 \leq r \leq R$, are built by relaxing the integrality restriction on a subset of variables. An iteration of the R&F algorithm consists in solving a subproblem, using B&B technique. From the obtained solution, another subset of integer variables are selected to be fixed to the values found during the following iterations. The key element is thus to appropriately define the above mentioned subsets of variables.

In the formulation from Paquay, Schyns, and Limbourg (2016), families X_{ij}, Y_{ij}, and Z_{ij} are real variables while all other families hold integer variables. If all integer variables are relaxed, the solution obtained can be too far from the original problem and the heuristic could be unable to compute a solution for the original MIP. Therefore, we define a *block*, denoted $\mathcal{B}$, as a selection of families of integer variables which is not affected by the integrality restriction relaxation. The block will be the basis of the subproblems construction. There exist many potential blocks. Only the most representative and intuitive blocks are considered in the present paper. The chosen blocks are described in Table 1. The set of families of variables is thus split into two parts: $\mathcal{B}$ contains the families in the selected block and $\neg\mathcal{B}$ contains all the other families.
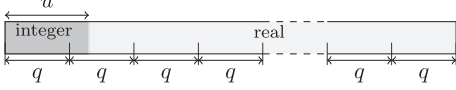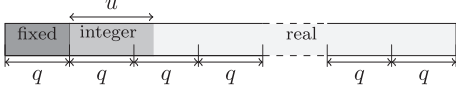
The construction of the subproblems MIP^r is box-oriented: at each iteration, the variables in $\mathcal{B}$ related to a subset of boxes are integer, while the variables in $\mathcal{B}$ related to the remaining boxes have the integrality constraint relaxed. The types (real or integer) of all the variables in $\neg\mathcal{B}$ are not changed during the whole algorithm. After solving MIP^r, we keep the values of the integer variables in $\mathcal{B}$ relative to a smaller subset of boxes, and fix them for the next iteration. In order to describe the subsets of boxes, two integers q and u, with $q < u$, are introduced. Parameter u denotes the number of boxes whose variables are integer and to be determined, and q the number of boxes whose variables values are kept. Because the biggest boxes are often difficult to pack, the sequence of boxes is first sorted by decreasing volume. At step r, the variables from $\mathcal{B}$ relative to the boxes from 1 to $(r-1) \times q$ are fixed to a previous value, the variables from $\mathcal{B}$ relative to the u next boxes (i.e. from $(r-1) \times q + 1$ to $(r-1) \times q + u$) are integer, and the variables from $\mathcal{B}$ relative to the remaining boxes (i.e. from $(r-1) \times q + u + 1$ to n) have the integrality restriction relaxed. Values for all the variables in $\neg\mathcal{B}$ are to be determined and are not affected by the integrality restriction relaxation. The types of variables in $\mathcal{B}$ during the first two steps of the R&F heuristic can thus be represented as they are in Table 2. The stopping criterion is based on the number of steps. The algorithm stops as soon as $q \times r + u \geq n$, i.e. $r \geq \frac{n-u}{q}$ and thus $R = \left\lceil \frac{n-u}{q} \right\rceil$.

Therefore, MIP^R provides a heuristic solution for the original MIP unless there exists an iteration r for which MIP^r is infeasible, which means the heuristic has failed.

3.2 *Insert-and-Fix heuristic*

The Insert-and-Fix (I&F) methodology also uses an iterative procedure to build an initial solution. Introduced by Liberalino (2012), this algorithm is similarly based on the decomposition of a large scale MIP into smaller subproblems. However, the heuristics I&F and R&F are different in the construction of these subproblems. First, in the subproblems built by the I&F method, not all variables are considered at every iteration, and variables are added step by step along the algorithm. The second difference is integrality constraints are never relaxed. At each iteration, all the considered variables have their

Table 2.   Types of the variables during the first two steps of a Relax-and-Fix heuristic with the block *Coord*.

Variables	Types
$X_{ij}, Y_{ij}, Z_{ij}, \forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\}$	Real
$u_j, p_{ij}, r_{iab}, x_{ik}^p, y_{ik}^p, z_{ik}^p, g_i, h_{ik}, a_{ik}, m_{ik}, o_{ik}, s_{ik}, \gamma_i^1, \gamma_i^2, \eta_{ik}^1, \eta_{ik}^2, \eta_{ik}^3, \eta_{ik}^4,$	Integer
$\beta_{ik}^l, \forall i, k \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, m\}$	

Step 1

$x_1, y_1, z_1, x_1', y_1', z_1', \ldots, x_u, y_u, z_u, x_u', y_u', z_u'$	Integer
$x_{u+1}, y_{u+1}, z_{u+1}, x_{u+1}', y_{u+1}', z_{u+1}', \ldots, x_n, y_n, z_n, x_n', y_n', z_n'$	Real

Step 2

$x_1, y_1, z_1, x_1', y_1', z_1', \ldots, x_q, y_q, z_q, x_q', y_q', z_q'$	Fixed to previous value
$x_{q+1}, y_{q+1}, z_{q+1}, x_{q+1}', y_{q+1}', z_{q+1}', \ldots, x_{q+u}, y_{q+u}, z_{q+u}, x_{q+u}', y_{q+u}', z_{q+u}'$	Integer
$x_{q+u+1}, y_{q+u+1}, z_{q+u+1}, x_{q+u+1}', y_{q+u+1}', z_{q+u+1}', \ldots, x_n, y_n, z_n, x_n', y_n', z_n'$	Real

type, i.e. integer or real, unchanged; however, as in the R&F algorithm, at the end of each iteration the values of a subset of variables are kept to be fixed at the next step.

The construction of the subproblems MIP^r relies on the idea of inserting boxes, step by step, inside ULDs. For this reason, the family u_j, which does not depend on the boxes i, is fully considered along the whole algorithm. All other variables, dependent of the boxes, will be partitioned into subsets, and these subsets will be added step by step along the algorithm. These subsets hold variables associated to several boxes. With the notations introduced for the R&F heuristic, we could define block $\mathcal{B}$ as the set of all real and integer families except u_j, while family u_j is $\neg\mathcal{B}$. Only families in $\mathcal{B}$ are partitioned into subsets to be introduced iteratively. As for the R&F heuristic, two parameters u and q, with $q < u$, are introduced to describe the subsets of variables. Parameter u denotes the number of boxes whose variables are to be determined, and q the number of boxes whose **integer** variables are kept and fixed at the next iteration. At step r, integer variables $\mathcal{B}$ relative to boxes from 1 to $(r-1) \times q$ are fixed at their optimal values from the solution of the previous subproblem (MIP^{r-1}), and all variables in $\mathcal{B}$ relative to the next u boxes are considered. The integer variables from $\mathcal{B}$ during the first two steps of the I&F heuristic can thus be represented as they are in Table 3. As for the R&F heuristic, the I&F algorithm must stop as soon as $q \times r + u \geq n$, i.e. $R = \left\lceil \frac{n-u}{q} \right\rceil$.
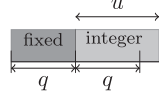
We need to be careful with the weight distribution constraints. Those are considered at each iteration for the boxes present at that phase of the algorithm. It is rather restrictive, but enforcing them all at once in the last step would tend to make the problem infeasible.

3.3 *Fractional Relax-and-Fix heuristic*

The Fractional Relax-and-Fix (FRF) methodology can be considered as a combination of the R&F and I&F heuristics: it is an iterative procedure working on a subset of the original MIP (as in the I&F), but also on the relaxation of the integrality constraints (as in the R&F). This methodology was introduced in Pochet and Wolsey (2006), and in Liberalino (2012). The basic concept is identical to the I&F heuristic, that is, only a subset of the original model is solved iteratively. Variables, and constraints inherent to them, are added step by step along the algorithm. The difference is the relaxation of the integrality restriction on part of the variables. The FRF heuristic is extended to suit the formulation developed in Paquay, Schyns, and Limbourg (2016), as follows.

As for the I&F, the subproblems correspond to a subset of boxes to be packed. For this reason, the set of variables is partitioned into two parts: $\neg\mathcal{B}$ contains family u_j, which is present in the whole algorithm, while $\mathcal{B}$ contains all the remaining families, which depend on the boxes. Families in $\mathcal{B}$ are decomposed into subsets which successively appear along the algorithm. The influence of the R&F arises among the families in $\mathcal{B}$. $\mathcal{B}$ contains families of real and integer variables.

Table 3. Existing variables during the first two steps of an Insert-and-Fix heuristic.

Variables	Types
$u_j, \quad \forall j \in \{1, \ldots, m\}$	Integer

Step 1

Variables	Types
$X_{ij}, Y_{ij}, Z_{ij}, \forall i \in \{1, \ldots, u\}, \forall j \in \{1, \ldots, m\}$	Real
$x_i, y_i, z_i, x'_i, y'_i, z'_i, a_{ik}, p_{ij}, r_{iab}, x^p_{ik}, y^p_{ik}, z^p_{ik}, g_i, h_{ik}, m_{ik}, o_{ik}, s_{ik}, \gamma^1_i, \gamma^2_i,$	Integer
$\eta^1_{ik}, \eta^2_{ik}, \eta^3_{ik}, \eta^4_{ik}, \beta^l_{ik}, \forall i, k \in \{1, \ldots, u\}, \forall j \in \{1, \ldots, m\}$	

Step 2

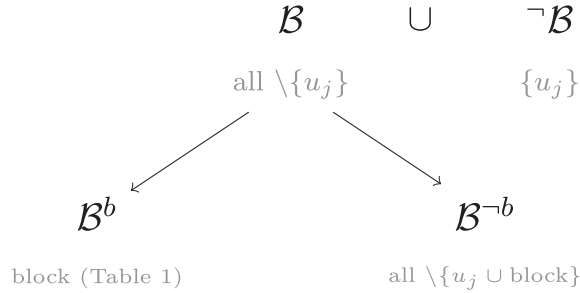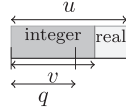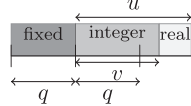| Variables	Types
$x_i, y_i, z_i, x'_i, y'_i, z'_i, a_{ik}, p_{ij}, r_{iab}, x^p_{ik}, y^p_{ik}, z^p_{ik}, g_i, h_{ik}, m_{ik}, o_{ik}, s_{ik}, \gamma^1_i, \gamma^2_i,$	Fixed to previous value
$\eta^1_{ik}, \eta^2_{ik}, \eta^3_{ik}, \eta^4_{ik}, \beta^l_{ik}, \forall i, k \in \{1, \ldots, q\}, \forall j \in \{1, \ldots, m\}$	
$X_{ij}, Y_{ij}, Z_{ij}, \forall i \in \{1, \ldots, q+u\}, \forall j \in \{1, \ldots, m\}$	Real
$x_i, y_i, z_i, x'_i, y'_i, z'_i, a_{ik}, p_{ij}, r_{iab}, x^p_{ik}, y^p_{ik}, z^p_{ik}, g_i, h_{ik}, m_{ik}, o_{ik}, s_{ik}, \gamma^1_i, \gamma^2_i,$	Integer
$\eta^1_{ik}, \eta^2_{ik}, \eta^3_{ik}, \eta^4_{ik}, \beta^l_{ik}, \forall i, k \in \{q+1, \ldots, q+u\}, \forall j \in \{1, \ldots, m\}$	

Figure 3. Variables partitioning for the FRF.

Among the integer families, several families are selected to have the integrality restriction partially relaxed. This selection of integer families, denoted $\mathcal{B}^b$, is a block as defined in the R&F heuristic. Blocks from Table 1 are also considered for this heuristic. Real and integer families in $\mathcal{B}$ that are not affected by the integrality restriction relaxation are denoted $\mathcal{B}^{\neg b}$. The structure can be represented as it is in Figure 3.

To describe the subsets of the families in $\mathcal{B}$, three integers q, v, and u, with $q < v < u$, are introduced. Parameter u denotes the number of boxes that are considered and whose variables are to be determined, v is the number of boxes, among the u boxes, whose variables have the integrality restriction to be satisfied, and q is the number of boxes whose variables in $\mathcal{B}$ are fixed. Parameter u affects all the families in $\mathcal{B}$, while parameters v and u only impact on the integer families in the block $\mathcal{B}^b$. At the first iteration, the idea is to insert the first u boxes of the sorted sequence, but relax some variables relative to the last $u - v$ boxes, and then keep the values of a selection of variables relative to the first q boxes. Specifically, we:

- introduce the variables relative to box 1 to box u from the families in $\mathcal{B}$;
- relax the integrality restriction on variables from the set $\mathcal{B}^b$ for the boxes $v + 1$ to u; and
- keep the values of the variables from the set $\mathcal{B}^b$ for the boxes 1 to q for the second iteration once the MIP is solved.

In the subproblem to be solved at iteration r, all the variables related to box 1 to box $(r-1) \times q + u$ are considered. Moreover, the integer variables from the families in block $\mathcal{B}^b$ related to box 1 to box $(r-1) \times q$ are fixed at their optimal values obtained when solving the previous subproblem (MIP^{r-1}), integer variables from the block $\mathcal{B}^b$ relative to boxes from $(r-1) \times q + 1$

Table 4. The first two steps of a Fractional Relax-and-Fix heuristic on block *Coord*.

Variables	Types
$u_j, \quad \forall j \in \{1, \ldots, m\}$	Integer
Step 1	
$X_{ij}, Y_{ij}, Z_{ij}, \forall i \in \{1, \ldots, u\}, \forall j \in \{1, \ldots, m\}$	Real
$p_{ij}, r_{iab}, x_{ik}^p, y_{ik}^p, z_{ik}^p, g_i, h_{ik}, a_{ik}, m_{ik}, o_{ik}, s_{ik}, \gamma_i^1, \gamma_i^2, \eta_{ik}^1, \eta_{ik}^2, \eta_{ik}^3, \eta_{ik}^4,$	Integer
$\beta_{ik}^l, \forall i, k \in \{1, \ldots, u\}, \forall j \in \{1, \ldots, m\}$	
$x_i, y_i, z_i, x_i', y_i', z_i', \forall i \in \{1, \ldots, v\}$	Integer
$x_i, y_i, z_i, x_i', y_i', z_i', \forall i \in \{v+1, \ldots, v+u\}$	Real
Step 2	
$x_i, y_i, z_i, x_i', y_i', z_i', \forall i \in \{1, \ldots, q\}$	Fixed to previous value
$X_{ij}, Y_{ij}, Z_{ij}, \forall i \in \{q+1, \ldots, q+u\}, \forall j \in \{1, \ldots, m\}$	Real
$p_{ij}, r_{iab}, x_{ik}^p, y_{ik}^p, z_{ik}^p, g_i, h_{ik}, a_{ik}, m_{ik}, o_{ik}, s_{ik}, \gamma_i^1, \gamma_i^2, \eta_{ik}^1, \eta_{ik}^2, \eta_{ik}^3, \eta_{ik}^4,$	Integer
$\beta_{ik}^l, \forall i, k \in \{1, \ldots, q+u\}, \forall j \in \{1, \ldots, m\}$	
$x_i, y_i, z_i, x_i', y_i', z_i', \forall i \in \{q+1, \ldots, q+v\}$	Integer
$x_i, y_i, z_i, x_i', y_i', z_i', \forall i \in \{q+v+1, \ldots, q+u\}$	Real

to $(r - 1) \times q + v$ are integer, but those related to boxes from $(r - 1) \times q + v + 1$ to $(r - 1) \times q + u$ have the integrality restriction relaxed. The first two steps of the FRF heuristic can thus be represented as they are in Table 4. The algorithm should stop as soon as $q \times r + v \geq n$, i.e. $R = \left\lceil \frac{n-v}{q} \right\rceil$.

Note that if $u = n$ then the FRF becomes an R&F heuristic, and if $v = u$ it becomes a particular case of I&F in which only a subset of variables relative to a box are fixed from one iteration to another.

3.4 *General remarks*

3.4.1 *Backtracking*

As in Baena, Castro, and González (2015), a backtrack process has been developed to overcome possible failures of the three presented matheuristics. Indeed, it may happen that a heuristic fails at one step, especially when the integrality restriction is removed and then reinserted. This does not mean the original problem is infeasible, and this failure does not bring any information about the suitability of this algorithm, except that some variables cannot be fixed to those values at that step. Backtracking works as follows: if a failure occurs at iteration r, the MIP^r is solved again but without fixing the integer variables related to the q previous boxes. If the heuristic fails again, then we backtrack once more, solving MIP^r without fixing the integer variables related to the $2q$ previous boxes. We can proceed that way $r - 1$ times, ending with just a relaxation of the initial MIP, which is supposed to be feasible.

3.4.2 *Additional ULDs*

One advantage of the matheuristics in comparison to a B&B resolution is how the number of ULDs is tackled. When using the B&B technique, the number of proposed ULDs is fixed and initially given. If this number is too large, the computational times explode, but considering too few ULDs may hinder the potential of a better loading solution. Since the heuristics now developed are iterative procedures, new ULDs can be added at each iteration.

Table 5. Information about the initial data set.

	Length [mm]	Width [mm]	Height [mm]	Weight [kg]	Volume [dm^3]	Density [kg/dm^3]
Range	[130;5250]	[100;1720]	[40;1900]	[1;1983]	[2.2;5832.96]	[0.01;5.62]
Average	927.65	620.18	618.89	106.55	571.57	0.24
Standard dev.	625.91	309.72	407.78	177.02	682.75	0.31

Notes: number of distinct boxes (types):199
Average number of boxes per type:2.82

Table 6. Best configurations found by `irace`.

	I&F				FRF			
Parameters	Types	Range	`irace` value	Parameters	Types	Range	Coord `irace` value	Coord & p_{ij} `irace` value
u	Integer	[2,7]	7	u	Integer	[3,7]	7	7
q	Integer	[1,3]	2	q	Integer	[2,6]	5	6
				v	Integer	[1,3]	1	1

4. Computational experiments

All tests were performed on a workstation with 32.0 GB RAM and an Intel Xeon processor E5-2620 v4 running 64-bit Windows 10 Pro. Codes were implemented in Java, and CPLEX 12.6 library was used as B&B solver. All the heuristics were run with a limited computational time of one hour.

Data sets used to perform the experimentations are available from `http://hdl.handle.net/2268/206856`. On the one hand, a set of six ULDs is used: three for the lower deck (LD1 with a cut 1, LD6 with a cut 1 and a cut 2, LD11 with no cut) and three for the main deck (PA with a cut 4, PG with a cut 3 and a cut 4, PM with no cut). More detail about these ULDs can be found in Boeing (2006). On the other hand, a box data set which stems from a real world case has been sampled to consider instances with different sizes in order to analyse the behaviour of the algorithms. The file originally contains information about the dimensions and weight for 562 rectangular boxes. The main features of these boxes are given in Table 5. For the specific rotations, if a box is too heavy ($>$50kg), it is assumed to be too much work to turn it, the parameters l^+, w^+ are thus assumed to be 0. For the stacking constraint, a box which has a density lower than 0.05 kg/dm^3 is considered as fragile. Data processing is used to generate box samples of different sizes ranging from 10 to 100 boxes. For each sample size, a set of 30 instances is built by random selection from the original data set. The training data sets were used for the parametrisation and the final data sets for the final experiments.

Some preliminary experiments have shown that the R&F heuristics are still too time consuming. They were not able to find feasible solutions in one hour of computational time for instances with 9 boxes. The same conclusion has been drawn for the block 'p_{ij}' and 'All i' of the FRF heuristics. Therefore, the heuristics I&F, FRF with the block 'Coord' and FRF with the block 'Coord & p_{ij}' will be trained and then analysed.

4.1 *Parametrisation with `irace`*

The heuristics I&F, FRF with the block 'Coord', and FRF with the block 'Coord & p_{ij}' have parameters that need to be tuned. Some experiments showed that computational times become very important for some parameter values, even for instances with small sizes. Therefore, a definite range was tested for each parameter, as shown in Table 6.

`irace` is a software package that provides an automatic configuration tool for tuning optimisation algorithms, that is, automatically finding the most appropriate settings of an algorithm given a set of instances of a problem, saving the effort that normally requires manual tuning (López-Ibáñez et al. 2016). During the tuning phase, a set of training instances representative of the problem (30 instances with 10, 15, 20, 25, and 30 boxes each) was provided to choose the best algorithm configuration. The selected algorithm configuration can then be used to solve new instances of the same problem. As explained in López-Ibáñez et al. (2016), along the racing method, if sufficient evidence is gathered that some candidate configurations perform statistically worse (validated with the non-parametric Friedman's statistical test) than at least one other configuration, such candidates are discarded and the procedure is iterated over the remaining surviving configurations. The elimination of inferior

Table 7. Average, minimum and maximum filling rates, average number of ULDs used in the solutions, number of unsolved instances, and average computational times for the three matheuristics and the B&B for different sample sizes (each sample size has 30 instances). Numbers in brackets for the B&B indicate the suboptimally solved instances.

	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$	$n = 70$	$n = 80$
Average ([min; max]) filling rate per ULD [%]								
I&F	41.63	39.60	43.02	46.25	45.36	43.16	46.23	45.68
	[13.1;62.6]	[11.5;64.7]	[1.3;70.2]	[9.9;79.7]	[2.8;72.3]	[6.9;71.0]	[8.3;72.4]	[1.4;71.9]
FRF with 'Coord'	41.49	43.07	45.39	50.90	48.91	49.89	49.78	51.09
	[13.11;62.6]	[11.9;62.6]	[5.0;69.0]	[19.0;82.1]	[4.3;70.9]	[13.9;73.5]	[21.1;69.9]	[24.28;72.45]
FRF with 'Coord & p_{ij}'	42.19	42.45	44.42	46.44	44.30	42.27	45.81	43.10
	[13.11;65.7]	[2.2;71.2]	[2.3;69.0]	[6.0;71.1]	[2.9;69.8]	[7.6;73.4]	[12.3;77.8]	[7.1;72.0]
Branch-and-Bound	41.39	29.63	–	–	–	–	–	–
	[7.2;62.6]	[16.2;45.5]						
Average number of ULDs used per instance								
I&F	1.50	1.90	2.10	3.57	4.61	5.83	6.96	8.04
FRF with 'Coord'	1.50	1.57	1.90	2.62	3.09	3.50	4.83	4.67
FRF with 'Coord & p_{ij}'	1.57	1.79	2.48	3.59	4.74	5.76	7.13	8.67
Branch-and-Bound	1.52	3	–	–	–	–	–	–
# unsolved instances								
I&F	0	0	1	2	2	1	2	2
FRF with 'Coord'	0	2	0	4	7	10	18	21
FRF with 'Coord & p_{ij}'	0	1	1	3	3	5	6	6
Branch-and-Bound	3(11)	24(6)	30(0)	30(0)	30(0)	30(0)	30(0)	30(0)
Average computational times [s.]								
I&F	12.73	106.88	41.48	116.14	114.70	155.35	264.91	317.07
FRF with 'Coord'	23.20	46.63	151.97	452.76	412.33	814.62	1603.66	2129.80
FRF with 'Coord & p_{ij}'	14.88	49.91	81.09	255.51	259.53	441.09	780.16	826.80
Branch-and-Bound	1935.43	3600.00	–	–	–	–	–	–

candidates speeds up the procedure and allows a more reliable evaluation of promising configurations. To assess the quality of a configuration, `irace` compares the objective function value of some configurations on different instances. Note that if a configuration does not find a feasible solution for one instance within one hour of computational time, then this configuration returns a large number, $n \times V_j$, to represent a poor quality solution. The best configurations found by `irace` are presented in Table 6.

One can observe that parameter u takes the upper limit of the range. However, the configuration with $u = 8$ is sometimes unable to find solutions for even medium size instances within one hour and is therefore discarded.

4.2 *Comparison of the parametrised heuristics*

In this section, the three parametrised matheuristics are tested on the final data sets, i.e. 30 instances with 10, 20, 30, …, 80 boxes each.

4.2.1 *Solution quality*

Among the 30 instances of each size, some heuristics may fail to find a feasible solution within one hour of computational time. In order to get a meaningful comparison, only the objective function values of the instances solved by the three methods are compared. The solution quality is thus compared in two steps, depending on the available results. The three methods are first compared on instances with a number of boxes ranging from 10 to 50 (Step 1). Then, the I&F and the FRF heuristic
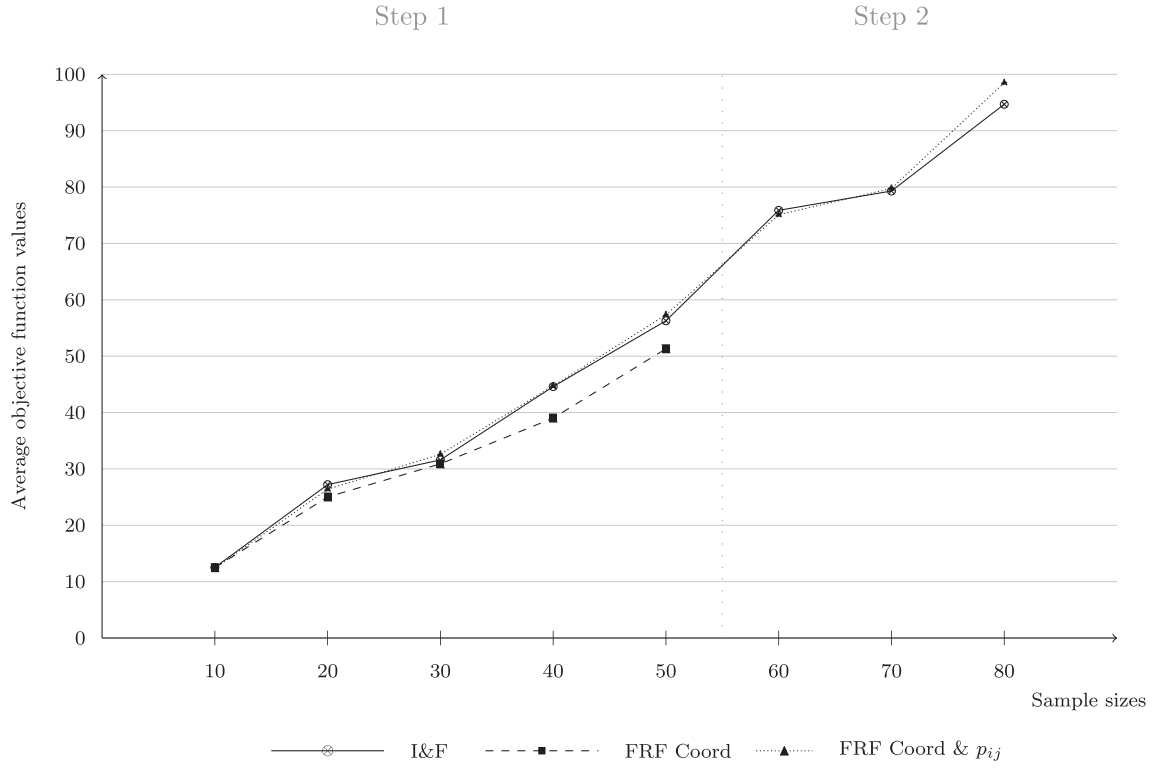
Figure 4. Comparison of the average objective function values obtained with the three heuristics for different sample sizes (each sample size has 30 instances).

with the block 'Coord & p_{ij}' are compared on instances with a number of boxes ranging from 60 to 80 (Step 2). The average objective function values computed over the instances solved by all the techniques (depending on the step) are plotted in Figure 4.

In order to compare the quality of the solutions provided by each method, Friedman's test and the post-hoc analysis using Conover test has been performed to measure if there is significant difference between the results. The FRF heuristic with the block 'Coord' significantly outperforms the two other techniques for instances with 20, 40, and 50 boxes. For instances with 80 boxes, the I&F heuristic significantly outperforms the FRF heuristic with 'Coord & p_{ij}'.

To get some insight into solution quality, the initial formulation was solved with a direct B&B application. Over the 30 instances with 10 boxes, the B&B failed for three instances, but those were solved by the three matheuristics. Thanks to the addition of ULDs, the FRF heuristics found better solutions than B&B for one instance. The heuristics I&F and FRF with 'Coord & p_{ij}' found better solutions than B&B in one hour for three instances. For the 16 instances optimally solved by the B&B, the average objective function value GAP is 4.5% for the I&F and 3.9% for the two FRF heuristics.

Table 7 provides the average, minimum and maximum filling rates and the average number of ULDs used in the solutions. These numbers are computed over the instances solved within one hour of computational time by the considered method, ignoring whether this instance was solved by the others. The matheuristics are apparently able to efficiently use the volume inside the ULDs. This is particularly remarkable given the maximum allowed centre of gravity height. One can observe that the FRF heuristic with the block 'Coord' has a higher average filling rate and a smaller average number of ULDs, in addition to the best average objective function value. Information about the B&B is provided for instances solved optimally or suboptimally within one hour of computational time.

4.2.2 *Computational times and unsolved instances*

The number of unsolved instances and the average computational times are presented in Table 7 for each sample size, for each method, and for the B&B. The average durations for each method were computed over the instances solved by the considered method, ignoring whether this instance was solved by the others. Regarding the B&B, an instance is considered

as unsolved if no feasible solution has been found within one hour. The number of instances suboptimally solved is indicated in brackets. The averages are computed over the instances optimally or suboptimally solved within one hour.

Looking at Table 7, one can see that the three matheuristics dramatically outperform the B&B with respect to computational time. The FRF heuristic with the block 'Coord' is the slowest heuristic. The second slowest technique is the other FRF heuristic. Finally, the I&F heuristic has more reasonable computational times and a smaller number of unsolved instances, even for large sample sizes.

4.2.3 *In practice*

For a reasonable number of boxes to be packed, the FRF heuristic with the block 'Coord' achieves good solution quality. However, for instances with more than 40 boxes the computational times become very large and the I&F heuristic becomes more interesting. Additional experiments would have to be run to clarify the precise size limit.

5. Conclusion

In this paper, the Multiple Bin Size Bin Packing Problem is addressed in three dimensions. The problem is extended to consider several additional constraints such as, cargo stability, fragility, weight distribution, and the special shapes of containers in air transportation. This specific MBSBPP has previously been formulated as an MIP, and our aim was to exploit this formulation by creating matheuristics able to find good initial solutions. We extended three techniques to suit this MIP: the heuristics Relax-and-Fix, Insert-and-Fix and Fractional Relax-and-Fix. The first algorithm was still very time consuming and has been put aside. The two remaining matheuristics have been tuned with the parametrisation tool called `irace`. Afterwards, they were tested on instances that were specially designed for this MBSBPP and the results were then analysed. We observed that two particular heuristics are promising because they quickly compute feasible solutions. Future research can be related to other sorting operators for the boxes. A method to speed up the process could be to halt the B&B search as soon as a feasible solution is obtained, or as soon as a given GAP is reached.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Baena, Daniel, Jordi Castro, and José A. González. 2015. "Fix-and-relax Approaches for Controlled Tabular Adjustment." *Computers & Operations Research* 58: 41–52.

Beraldi, Patrizia, Gianpaolo Ghiani, Antonio Grieco, and Emanuela Guerriero. 2008. "Rolling-horizon and Fix-and-relax Heuristics for the Parallel Machine Lot-sizing and Scheduling Problem with Sequence-dependent Set-up Costs." *Computers & Operations Research* 35 (11): 3644–3656.

Boeing. 2006. "Weight and Balance Control and Loading Manual – Sample Manual – Model 747–430."

Bortfeldt, A., and G. Wäscher. 2013. "Constraints in Container Loading – A State-of-the-art Review." *European Journal of Operational Research* 229: 1–20.

Brunetta, L., and P. Gregoire. 2005. "A General Purpose Algorithm for Three-dimensional Packing." *INFORMS Journal on Computing* 17 (3): 328–338.

Ceschia, S., and A. Schaerf. 2013. "Local Search for a Multi-drop Multi-container Loading Problem." *Journal of Heuristics* 19 (2): 275–294.

Chan, F. T. S., R. Bhagwat, N. Kumar, M. K. Tiwari, and P. Lam. 2006. "Development of a Decision Support System for Air-cargo Pallets Loading Problem : A Case Study." *Expert Systems with Applications* 31: 472–485.

Che, Chan Hou, Weili Huang, Andrew Lim, and Wenbin Zhu. 2011. "The Multiple Container Loading Cost Minimization Problem." *European Journal of Operational Research* 214 (3): 501–511.

Chen, C. S., S. M. Lee, and Q. S. Shen. 1995. "An Analytical Model for the Container Loading Problem." *European Journal of Operational Research* 80: 68–76.

Dowsland, William B. 1991. "Three-dimensional Packing – Solution Approaches and Heuristic Development." *International Journal of Production Research* 29 (8): 1673.

Faroe, Oluf, David Pisinger, and Martin Zachariasen. 2003. "Guided Local Search for the Three-dimensional Bin-packing Problem." *INFORMS Journal on Computing* 15 (3): 267.

Hemminki, J., T. Leipala, and O. Nevalainen. 1998. "On-line Packing with Boxes of Different Sizes." *International journal of production research* 36 (8): 2225–2245.

Kelly, Jeffrey D., and John L. Mann. 2004. "Flowsheet Decomposition Heuristic for Scheduling: A Relax-and-fix Method." *Computers & Chemical Engineering* 28 (11): 2193–2200.

Liberalino, H. 2012. "Problèmes de Production avec Transport des Composants." PhD diss., Université Blaise Pascal – Clermont-Ferrand II.

Limbourg, S., M. Schyns, and G. Laporte. 2012. "Automatic Aircraft Cargo Load Planning." *Journal of the Operational Research Society* 63: 1271–1283.

Lin, Jin-Ling, Chir-Ho Chang, and Jia-Yan Yang. 2006. "A Study of Optimal System for Multiple-Constraint Multiple-container Packing Problems." In *Advances in Applied Artificial Intelligence*. Vol. 4031, Lecture Notes in Computer Science, edited by Moonis Ali, Richard Dapoigny, 1200–1210. Berlin Heidelberg: Springer.

López-Ibáñez, Manuel, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. 2016. "The Irace Package: Iterated Racing for Automatic Algorithm Configuration." *Operations Research Perspectives* 3: 43–58.

Ngoi, B. K. A., M. L. Tay, and E. S. Chua. 1994. "Applying Spatial Representation Techniques to the Container Packing Problem." *International Journal of Production Research* 32 (1): 111–123.

Oliveira, Beatriz Brito, Maria Antonia Carravilla, José Fernando Oliveira, and Franklina M. B. Toledo. 2014. "A Relax-and-fix-based Algorithm for the Vehicle-reservation Assignment Problem in a Car Rental Company." *European Journal of Operational Research* 237 (2): 729–737.

Paquay, Célia, Michael Schyns, and Sabine Limbourg. 2016. "A Mixed Integer Programming Formulation for the Three-dimensional Bin Packing Problem Deriving from an Air Cargo Application." *International Transactions in Operational Research* 23 (1–2): 187–213.

Pochet, Yves, and Laurence A. Wolsey. 2006. *Production Planning by Mixed Integer Programming*. New York: Springer.

Wäscher, G., H. Haußner, and H. Schumann. 2007. "An Improved Typology of Cutting and Packing Problems." *European Journal of Operational Research* 183: 1109–1130.