The Three-Dimensional Knapsack Problem with Balancing Constraints

Mauro Baldi[a], Guido Perboli[a,b,*], Roberto Tadei[a]

[a] Politecnico di Torino, Turin, Italy
[b] CIRRELT, Montreal, Canada

Abstract

In this paper we introduce a new Packing problem, the Three-Dimensional Knapsack Problem with Balancing Constraints (3BKP), the extension of the Three-Dimensional Knapsack Problem (3KP) where additional constraints related to the packing center of mass are given. The 3BKP consists in orthogonally packing a subset of three-dimensional weighted items into a knapsack in order to maximize the total profit of the loaded items. The items must not overlap and the packing center of mass must lie into a predefined boxed domain inside the knapsack. We assume that items can be rotated. We give a MIP model for the problem, upper bounds and an efficient heuristic to solve large size instances. The computational results show that the MIP model cannot find optimal solutions, except for small size instances, but it can be used to calculate upper and lower bounds. It is shown that our heuristic outperforms the solution quality both of the MIP model and the heuristics available in the literature explicitly designed to solve 3KP.

Keywords: 3D Knapsack, load balancing, MIP model, heuristic

1. Introduction

A major challenge in the loading problem is taking into account load balancing constraints. These kind of constraints arise in many practical applications such as aircraft loading [1], space cargo loading [2], and maritime transportation [3]. The balancing issue is extremely important in some very risky applications, e.g. space cargo loading, but it is of great interest also in other applications, e.g. air and maritime cargo loading, where safety and cost of loading issues must be considered. From the cost viewpoint, it has been shown (Mongeau and Bés [4]) that minor displacements from an ideal center of mass can result in a significant increasing of fuel consumption for aircrafts and ships. Despite its importance, the issue of the balance loading has not been deeply studied. This is mainly due to the difficulty of extending exact and heuristic methods developed for the multi-dimensional packing to the balanced case. In fact, most of these methods use geometric properties in order to reduce the computational effort and these properties do not hold anymore when the balancing constraints are considered.

The aim of this paper is threefold. First, we introduce a new Packing problem, the Three-Dimensional Knapsack Problem with Balancing Constraints, the extension of the Three-Dimensional Knapsack Problem (3KP) where additional constraints related to the packing center of mass are given.

Second, we give a MIP model for the problem. Computational experience shows that the model, except for small size instances, cannot solve to optimality the problem. Nevertheless, it can be used to calculate upper and lower bounds.

Finally, we introduce 3BKP-H, a new heuristic which generalizes the existing heuristic UniPack, developed for multi-dimensional Packing (Crainic et al. [5] and Perboli et al. [6]), and the Extreme Point rule (Crainic et al. [7]) for the items accommodation, in order to cope with the center of mass constraints.

We want to point out that balancing conditions are not equivalent to stability conditions. In particular, balancing conditions explicitly keep the packing center of mass into account, i.e. it must lie within a given three dimensional

*Corresponding author.
Department of Control and Computer Engineering, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino (Italy)
Tel. +39 0110907097, E-mail: guido.perboli@polito.it

domain. Stability conditions are related to the definition of indices related to the center of mass, but do not explicitly consider it (see [3] for further details). The most popular stability conditions are related to the supporting surface (i.e. the percentage of area on which an item lays down) and the number of items making up the supporting surface itself. The latter is related to the geometrical stability of the item and is a measure of the bottom surface which prevents the item to fall down. Another condition is related to the number of supporting items. In fact, practitioners prefer that the supporting surface is made by more than one item, in order to avoid that the movement of a single item can cause the upper item to collapse. Another index of stability (see, for instance, [8]) is given by the maximum weight allowed along the vertical axis. This index is useful to take into account the fragility of certain items.

New test instances are introduced and used to derive extensive computational results. It is shown that our heuristic outperforms the solution quality both of the MIP model and the heuristics available in the literature which are explicitly designed to solve 3KP.

The remainder of the paper is organized as follows. In section 2 we introduce the problem and the MIP model. In section 3 a state of the art is presented by considering 3D Packing problems, 3D Knapsack problems and, eventually, 3D Knapsack problems with balancing constraints. Section 4 is devoted to exact methods to solve the problem and to derive upper and lower bounds. Section 5 introduces our heuristic 3BKP-H. In section 6 extensive computational results are given. Finally, the conclusions of our work are reported in section 7.

2. Problem description and MIP model

The 3BKP is defined as follows: given a knapsack K with size W (width), D (depth), and H (height), and a set of items $J = \{1, ..., n\}$ with profit p_j, size w_j (width), d_j (depth), and h_j (height), and mass m_j, we want to assign a subset of items $J' \subseteq J$ to the knapsack K such that J' is a feasible loading for the knapsack itself, and the total profit of the loaded items is maximum. We denote k the cardinality of J', i.e. the number of loaded items. Feasibility requires that loaded items do not overlap and the overall center of mass position is inside a given three-dimensional domain. For three-dimensional domain we mean any three-dimensional connected set of points within the knapsack. In this paper we treat parallelepiped domains, given by the following cartesian product: $\left[L^1, U^1\right] \times \left[L^2, U^2\right] \times \left[L^3, U^3\right]$, where, as we will show in Section 2.1, L^δ and U^δ are the coordinates of the vertices of the parallelepiped along dimension δ. Figure 1 shows an example of a three-dimensional domain with its projections on the (x, y), (x, z), and (y, z) plans. Following Wäscher et al. classification [9], 3BKP is a Three-Dimensional Single Large Object Placement Problem (3D-SLOPP) with balancing constraints (3DB-SLOPP).

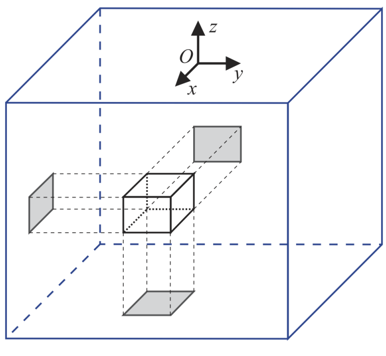

Figure 1: A three dimensional domain

Furthermore the following assumptions are made:

- the items and the knapsack have parallelepiped shape
- the origin of the knapsack and of each item is located at their own bottom-left-back corner (see Figure 2)
- the knapsack is located in the first octant of the 3D Cartesian coordinate system, with its origin placed in position (0, 0, 0) (see Figure 2)

- items can rotate so that each item side is parallel to one axis
- knapsack and items walls have negligible thickness; this means that an item can be placed at the very end of another item. Negligible thinkness of the knapsack walls means that the whole knapsack volume WDH can be exploited. If, for instance, the knapsack walls had thikness ϵ, then the available volume would be $(W - 2\epsilon)(D - 2\epsilon)(H - 2\epsilon)$
- knapsack and items size is assumed to be a non-negative integer.

Considering the accommodated items, the value of the overall profit P can be calculated as:

$$P = \sum_{j \in J'} p_j. \tag{1}$$

Note that high or low values of P do not necessarily correspond to high or low knapsack volume exploitations because, in principle, there is no correlation among volumes and profits of items. This also mean that a high profit item not necessarily has a high volume. There can be items with small volumes but high profits.

We define $\vec{r}_{CM_j}$ the radius vector (position) of the center of mass of any accommodated item j, and $M = \sum_{j \in J'} m_j$ the overall mass of the accommodated items. The packing center of mass position is given by the average of the loaded items positions weighted by their masses m_j:

$$\vec{r}_{CM} = \sum_{j \in J'} \vec{r}_{CM_j} m_j / M. \tag{2}$$

Ideally, we would like the packing center of mass lie at a given point. This is impossible in practice and so we relax this condition requiring the overall center of mass to be within a given three dimensional domain. If, as a measure of solution quality, we used the distance between the packing center of mass and the ideal center of mass, then, all the solutions with their center of mass lying on a sphere with center the ideal center of mass would be equivalent. Nevertheless, in real life situations, packings with their center of mass toward the bottom of the knapsack are more desirable. To take this issue into account we use the unbalancing index U, which relies on the standard deviation, as figure of merit for evaluating solutions quality.

The unbalancing index U is then a measure of the dispersion of the actual packing center of mass $\vec{r}_{CM}$ with respect to an ideal position $\vec{r}^{\,'}_{CM}$.

To define U, we first consider the standard deviation of a set of values x_j, with $j \in J'$ and arithmetic mean $\bar{x}$, defined as follows:

$$\sigma_x = \sqrt{\frac{\sum_{j \in J'} (x_j - \bar{x})^2}{k}}. \tag{3}$$

The formula for the unbalancing index U is obtained by plugging in (3) the radius vectors of the centers of mass. Moreover, since we are dealing with vectors, the modulus of their difference must be considered:

$$U = \sqrt{\frac{\sum_{j \in J'} \left|\vec{r}_{CM_j} - \vec{r}^{\,'}_{CM}\right|^2}{k}}. \tag{4}$$

More details can be found in [2].

2.1. The model

Let us define:

- J: the set of items, with cardinality n and associated indexes i and j
- Δ: the set of dimensions $\{1, 2, 3\}$, with associated index δ
- R: the set of rotations, with cardinality 6 and associated index r

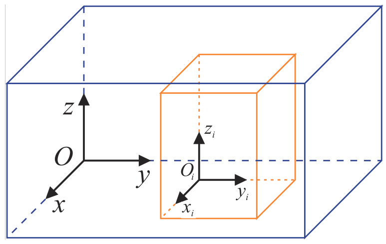

Figure 2: knapsack and item placement

- s_{ir}^{δ}: the size of item i along dimension δ when the item is rotated with rotation r
- S^{δ}: the knapsack size along dimension δ; in particular $\left(S^1, S^2, S^3\right) = (W, D, H)$
- L^{δ}, U^{δ}: lower and upper bounds along dimension δ which limit the domain where the packing center of mass must lie within
- γ_{ir}^{δ}: the coordinate of the center of mass of item i along dimension δ when the item is rotated with rotation r. This coordinate is calculated with respect to the bottom-left-back point of the item
- χ_i^{δ}: the coordinate of the bottom-left-back point of item i along dimension δ
- t_i: a binary variable which assumes value 1 if item i is loaded into the knapsack, 0 otherwise
- b_{ij}^{δ}: a binary variable which assumes value 1 if item i comes *before* item j along dimension δ, 0 otherwise. Item i comes before item j along dimension δ if $\chi_i^{\delta} < \chi_j^{\delta}$. Since items cannot overlap, we can be more restrictive on such inequation by imposing that $\chi_i^{\delta} + s_{ir}^{\delta} \leq \chi_j^{\delta}$ if item i is rotated with rotation r
- ρ_{ir}: a binary variable which assumes value 1 if item i is rotated with rotation r, 0 otherwise.

Finally, the notation $i < j$ means that index i precedes index j in the given items from 1 to n.

The model for the Three-Dimensional Knapsack Problem with Balancing Constraints (3BKP-M) can then be formulated as follows:

$$\max \sum_{j \in J} p_j t_j \tag{5}$$

$$\text{s.t.} \quad \sum_{j \in J} w_j d_j h_j t_j \leq WDH \tag{6}$$

$$\sum_{\delta \in \Delta} (b_{ij}^{\delta} + b_{ji}^{\delta}) \geq t_i + t_j - 1, \quad i < j,\ i \in J,\ j \in J \tag{7}$$

$$\chi_i^{\delta} + \sum_{r \in R} s_{ir}^{\delta} \rho_{ir} \leq S^{\delta} \quad i \in J,\ \delta \in \Delta \tag{8}$$

$$\chi_i^{\delta} + \sum_{r \in R} s_{ir}^{\delta} \rho_{ir} \leq \chi_j^{\delta} + M(1 - b_{ij}^{\delta}), \quad i < j,\ i \in J,\ j \in J,\ \delta \in \Delta \tag{9}$$

$$\chi_j^{\delta} + \sum_{r \in R} s_{jr}^{\delta} \rho_{ir} \leq \chi_i^{\delta} + M(1 - b_{ji}^{\delta}), \quad i < j,\ i \in J,\ j \in J,\ \delta \in \Delta \tag{10}$$

$$\chi_i^{\delta} \leq M t_i \quad i \in J,\ \delta \in \Delta \tag{11}$$

$$b^{\delta}_{ij} \leq t_i \quad i \in J,\ j \in J,\ \delta \in \Delta \tag{12}$$

$$b^{\delta}_{ji} \leq t_j \quad i \in J,\ j \in J,\ \delta \in \Delta \tag{13}$$

$$\sum_{i \in J} m_i \chi^{\delta}_i + \sum_{i \in J} \sum_{r \in R} m_i \gamma^{\delta}_{ir} \rho_{ir} \geq L^{\delta} \sum_{i \in J} m_i t_i \quad \delta \in \Delta \tag{14}$$

$$\sum_{i \in J} m_i \chi^{\delta}_i + \sum_{i \in J} \sum_{r \in R} m_i \gamma^{\delta}_{ir} \rho_{ir} \leq U^{\delta} \sum_{i \in J} m_i t_i \quad \delta \in \Delta \tag{15}$$

$$\sum_{r \in R} \rho_{ir} = 1, \qquad i \in J \tag{16}$$

$$\chi^{\delta}_i \geq 0 \quad i \in J,\ \delta \in \Delta \tag{17}$$

$$t_i \in \{0, 1\} \quad i \in J \tag{18}$$

$$b^{\delta}_{ij} \in \{0, 1\} \quad i \in J,\ j \in J,\ \delta \in \Delta \tag{19}$$

$$\rho_{ir} \in \{0, 1\} \quad i \in J,\ r \in R. \tag{20}$$

The objective function (5) gives the total profit of the selected items. Constraint (6) expresses the capacity constraints, i.e. the sum of the volumes of the selected items must not exceed the knapsack volume. It is redundant with constraints (8), but we keep it because it helps the solver providing better continuous relaxations in some instances. Constraints (7) ensures that two packed items do not overlap. Constraints (8) state that items must lie inside the knapsack, i.e. for each dimension δ the sum of the coordinate of the bottom-left point with the dimension of the item must give a value less or equal than the size of the knapsack along dimension δ. Constraints (9) state that, if item i comes *before* item j, then the sum of the position of item i plus its size must be less or equal than the position value of item j along dimension δ. Constraints (10) have the same meaning, this time with item j coming before item i. Note that constraints (7) - (10) are taken from [10, 11, 12]. Constraints (11) express that, if item i is not selected, then its placement coordinates must be zero. A similar meaning have constraints (12) and (13) that state that, if an item is not selected, then it cannot be placed before another one. Constraints (14) and (15) ensure balancing conditions and can be derived from the center of mass definition (2). Note, in fact, that the overall center of mass position along dimension δ is given by

$$r^{\delta}_{CM} = \frac{\sum_{i \in J} m_i \chi^{\delta}_i + \sum_{i \in J} \sum_{r \in R} m_i \gamma^{\delta}_{ir} \rho_{ir}}{\sum_{i \in J} m_i t_i}.$$

Since balancing constraints imply that the overall center of mass along dimension δ must lie within L^{δ} and U^{δ}, i.e. $L^{\delta} \leq r^{\delta}_{CM} \leq U^{\delta}$ then constraints (14) and (15) hold. Equations (16) imply that each item must be rotated according exactly one of the six available rotations. Finally, the involved variables domains follow.

3. State of the art

The 3BKP is a problem belonging to the Cutting and Packing (C&P) family. Wäscher et al. [9] have recently published a classification for C&P problems which extends an older one due to Dyckhoff [13]. According to Wäscher et al classification, the 3BKP is a Three-Dimensional Single Large Object Placement Problem (3D-SLOPP) with balancing constraints (3DB-SLOPP) [9]. In the following we present the literature along two main components of 3BKP: multidimensional packing and balancing constraints.

A first attempt to model multidimensional packing was due to Gilmore and Gomory [14] dealing with column generation. Baldacci and Boschetti [15] developed a Branch and Cut for solving the problem. Other contributions come from Beasley [16], Chung et al. [17], Berkey and Wang [18], and Fekete and Schepers [19].

Martello et al. [20] introduced the concept of Corner Points. Extensions of their work can be found in den Boef et al. [21], Martello et al. [22], and Crainic et al. [7]. In particular, Crainic et al. [7] introduced an extension of the Corner Points, the Extreme Points. Being the basis of the heuristic introduced in this paper, the Extreme Points will be discussed in detail in Section 5. Some applications work with a different geometry. One of the most studied is the case where the items have a spherical shape rather than a parallelepiped one. That is the case of the sphere packing. Musès [23] tackles the problem in multiple dimensions.

The specific literature on multidimensional Knapsack packing problems is huge (see [24] for a recent survey). In the following we will give only some on the main references. Papers tackling this problem are Hadjiconstantinou and Christofides [25], Boschetti et al. [26], George and Robinson [27], and Fekete and Schepers [28, 29]. Other contributions are due to Caprara and Monaci [30] and Clautiaux et al. [31] which deal with exact approaches to the multidimensional knapsack problems. Multidimensional Knapsack problems can be further extended to the case of multiple objective functions. An interesting work is due to Mavrotas et al. [32]. To the best of our knowledge the latest contribution to 3D Knapsack problems comes from Egeblad and Pisinger [33], where the authors propose an exact model and heuristics for 2D and 3D Knapsack problems. Unfortunately, their model is useless to derive both lower and upper bounds, whilst their heuristic manages instances up to 60 items for the 3D case.

To the authors knowledge there are only few papers on packing problems dealing with non-linear or balancing constraints. Various approaches, including artificial intelligence or simulated annealing have been considered to tackle different cargo issues (see [34], [35], [2], and [36]) and are related to aircraft loading problems. A MIP model for the 3BKP can be found in Fasano [37], where additional equations to meet balancing conditions are taken from Williams [38]. MartinVega [39] focused his research on splitting the set of items into groups to be assigned to different airplanes, without considering the packing problem. Cochard and Yost [40] developed a heuristic that first solves the packing problem and then tries to balance the airplane by swapping groups of items. The most relevant works on loading with balancing constraints are the ones led by Amiouny et al. [41] and Mathur [42]. In both papers the authors investigate on the accommodation of preloaded containers in fixed positions, with balancing constraints to be satisfied in one dimension only. Colaneri et al. [2] presented a MIP-based heuristic to solve a specific 3D packing problem related to the space cargo loading, but the specific packing constraints make hard to use their model for the item accommodation in 3BKP. An application more similar to 3BKP can be found in Kaluzny and Shaw [1], where a variant of the 3D packing problem is introduced, but the balancing is not considered as a constraint. In fact, the authors use the balancing in the objective function, minimizing its deviation from a specific point. Moreover, they solved their instances by means of a MIP model, which makes their approach impracticable even with 20 items.

4. Exact methods and upper bounds

In the following, we will discuss why it is not possible to extend the available exact methods for multi-dimensional packing problems to 3BKP and how upper bounds for this problem can be computed.

Most of the exact methods which are effective for standard multi-dimensional packing problems try to reduce the number of the possible locations of items to be added to an existing packing, as for the *Corner Points* by Martello et al. [20], or the number of possible packings by means of an implicit representation of classes of equivalent packings, as in the *Packing Class* approach by Fekete and Schepers [28]. Unfortunately, both approaches cannot be extended to the balanced case. While Corner Points push the items towards a corner of the knapsack, which clearly makes impossible to represent some optimal solutions of the balanced case, the representation used in Packing Class loses its main advantage of collapsing several (potentially exponential) packings in the same Packing Class when balancing constraints are present. In fact, Packing Class works on the idea that, given two items which are placed in a packing, one after the other, by swapping their mutual order the packing volume usage does not change. Unfortunately, whilst this is true from a geometrical point of view, it does not work from a balancing point of view.

Thus, in order to solve to optimality the 3BKP, the only current available approach is through the MIP model 3BKP-M, given by (5)-(20). 3BKP-M, even if implemented by the most efficient commercial solvers, is not able to solve instances with more than 20 items. This is mainly due to the poor quality of its continuous relaxation. The 3BKP-M relaxation could be strengthened by means of cut generation techniques as in [1]. Unfortunately, our computational experience has shown that these cuts are not effective when the number of items becomes larger than 20. Nevertheless, the MIP model can be profitably used to calculate upper and lower bounds for 3BKP, as we will show in Section 6.

A first upper bound, named UB_{1D}, can be calculated by solving a one-dimensional knapsack problem, i.e. the model 3BKP-M where the constraints (7)-(16) are ignored. The relaxed problem becomes a one-dimensional knapsack problem where the knapsack maximum weight is WDH and the weight of each item j is $w_j d_j h_j$. This approach has been used in literature by Fekete and Schepers [19], Caprara and Monaci [30] and Baldacci and Boschetti [15].

In principle, this bound could be strengthened by means of dual feasible functions ([28, 15]), which can be applied both to the oriented and non oriented case. Unfortunately, the test we performed on the model shows that known dual feasible functions are not effective when the balancing constraints hold.

A second upper bound, named UB_{RM}, could be obtained from the linear relaxation of model (5)-(20). Previous tests on similar models for the unbalanced case show that this bound has the same quality of UB_{1D} ([37, 33]). In the following UB_{1D} will then be used.

5. The heuristic

It is trivial to show that the 3BKP is NP-Hard, being an extension of the Three-Dimensional Knapsack Problem, which is NP-Hard [33]. In the following, we present 3BKP-H, an efficient heuristic conceived to solve 3BKP, which is an extension of the UniPack framework by Crainic et al. [5] and Perboli et al. [6]. As stated by Fasano [37], the MIP model is hard to solve using standard techniques, while we showed in Section 4 how other properties used to compute exact solutions for standard multi-dimensional packing problems are not valid for the 3BKP. Thus, that justifies a heuristic approach to solve the 3BKP.

UniPack is a heuristic able to solve many packing problems which differ in the objective function and constraints. It is based on the concept of Extreme Points (*EP*s), introduced by Crainic et al. [7]. These are a further extension of the Corner Points introduced by Martello et al. [20].

Corner Points are the non-dominated locations where an item can be placed into an existing packing. In two dimensions, Corner Points are defined where the envelope of the items in the knapsack changes from vertical to horizontal (the green dots in Figure 3).

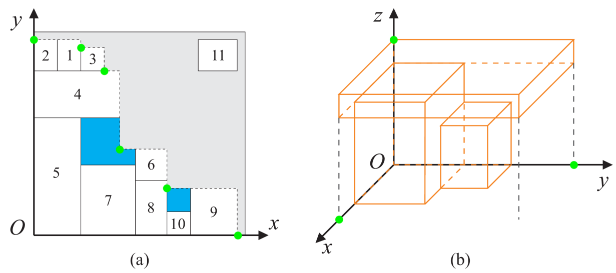

Figure 3: Corner Points in 2D and 3D packings

Heuristics using Corner Points can be inefficient in terms of knapsack utilization. Consider, for example, the packing depicted in Figure 3 and item 11. According to the Corner Points definition, one can add the item on any of the green dots. It is clear, however, that item 11 could also be placed into one of the shaded regions, which the Corner Points do not allow to exploit.

Extreme Points (*EP*s) provide the means to exploit the free space defined inside a packing by the shapes of the items already in the knapsack. Figure 4 illustrates *EP*s in 2D and 3D packings.

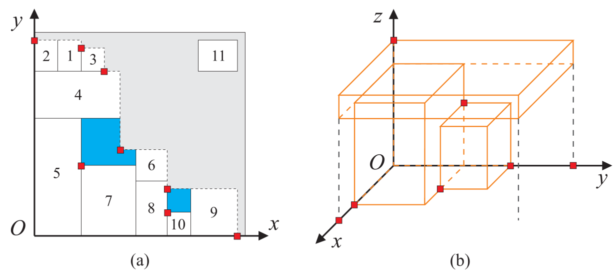

Figure 4: Extreme Points in 2D and 3D packings

The basic idea of the *EP*s is that when an item j with sizes (w_j, d_j, h_j) is added to a given packing and is placed with its left-back-down corner in position (x_j, y_j, z_j), it generates a series of new potential points, the *EP*s, where additional items can be accommodated. The new *EP*s are generated by projecting the points with coordinates $(x_j + w_j, y_j, z_j)$, (x_j, y_j+d_j, z_j), and (x_j, y_j, z_j+h_j) on the orthogonal axes of the knapsack. Figure 5 illustrates the concept.

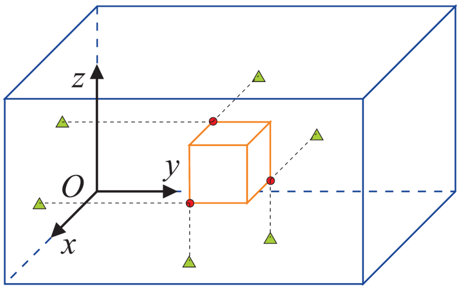

Figure 5: Extreme Points (the triangles) defined by an item

Beside the saving of space by applying Extreme Points rather than Corner Points, another advantage is the time complexity to find an extreme point set. As proved in Crainic et al. [7], the overall computational effort is $O(n)$- where n is the number of items- whilst Corner Points require a $O(n^2)$ complexity.

The general scheme of the heuristic 3BKP-H is depicted in Figure 6.

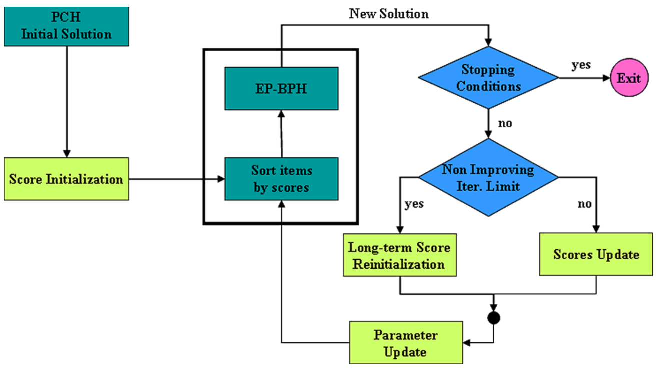

Figure 6: General scheme of 3BKP-H

The core of 3BKP-H is an accommodation procedure, the *EP*-based constructive heuristic, named EP-BPH. 3BKP-H is also used to build the initial solution by applying several sorting criteria to the items and retaining the best one as the initial solution. In the following, we refer to this composite heuristic as PCH (Packing Constructive Heuristic).

We assign a score to each item, thus specifying the order in which items are to be considered by the accommodation heuristic. The score definition is problem specific.

Scores are thus first initialized through the *Score Initialization* procedure, and then are dynamically modified by means of the *Score Update* and *Long-term Score Reinitialization* procedures. *Score Update* proceeds through small changes, aiming to adjust the scores used to sort the items at iteration k of 3BKP-H according to the quality of the solution built at iteration $k-1$. *Long-term Score Reinitialization* incorporates long-term decisions, as long-term memory structures, and proceeds through larger score modifications in the scores in order to avoid cycling on the same solutions and explore new regions of the solution space.

Score computation and update depend upon a number of parameters. We aim to keep this number as lower as possible to simplify their adjustment during computation. 3BKP-H provides a dynamically-adjusting parameter procedure denoted *Parameter Update* (see section 5.3).

The main steps on 3BKP-H are the following:

- Build an initial solution of the packing problem and set the best-solution BS equal to the initial solution. We use the PCH heuristic
- Scoring Phase
  - Initialize the score of the items: the *Score Initialization* procedure
  - While *Stopping Conditions* are not encountered, repeat the following steps:
    - Sort the items according to their scores and apply a constructive heuristic to the sorted list, obtaining a new solution CS. We use the EP-BPH procedure
    - If a given number of successive non-improving iterations is reached, reinitialize the scoring using the *Long-term Score Reinitialization* procedure; otherwise, update the scores using the *Score Update* procedure according to the CS solution
    - If CS is better than BS, then set BS to CS
    - The *Parameter Update* procedure then internally adjusts the parameters.

5.1. EP*-based Constructive Heuristics for Non-Guillotine Orthogonal Higher-Dimensional Packing Problems*

We now present the constructive heuristic PCH and the initial solution procedure EP-BPH we propose for Non-Guillotine Orthogonal Higher-Dimensional Packing problems. The procedures are based on the *Best Fit Decreasing* (BFD) idea and generalize the heuristic presented in Crainic et al. [7]. An interesting work on this topic is the one of Burke et al. [43, 44].

Following an initial sorting of the items by non-increasing order of their volumes, the BFD constructive heuristic for 1D Bin Packing problem tries to load each item into the best bin. The latter is defined as the bin which, after loading the item, has the minimum free volume, defined as the bin volume minus the sum of the volumes of the items it contains. If the item cannot be accommodated into the existing bins, a new bin is created . Despite its simplicity, the BFD heuristic offers good performances for 1D Bin Packing problems. Similar heuristics exist for other packing problems, e.g. Knapsack and Strip Packing. Unfortunately, extending these heuristics to a general constructive heuristics for Non-Guillotine Orthogonal Higher-Dimensional Packing problems is a non-trivial task. On the one hand, while in 1D cases the ordering is done considering a unique attribute characterizing both items and bins, i.e. their volume or profit, more choices exist in the multi-dimensional context. One may thus consider sorting items according to their width, depth, or height, as well as, derived from these attributes, according to their volume or the areas of their different faces. Consequently, best bin definition in the BFD heuristic is not unique. On the other hand, whilst items accommodation does not need to be taken into account in 1D problems, a 2D or 3D packing may significantly vary according to how items are placed inside the bins, even when their ordering and the rule for selecting the best bin are not changed. Moreover, according to the packing problem, the number of available bins may be unlimited or fixed and all the items or just a subset of them must be loaded.

We propose a new constructive heuristic based on BFD ideas, denoted *Extreme-Point Best Positioning Heuristic* (EP-BPH), which places the items into bins using the Extreme Points concept. As indicated earlier, the Extreme Points define the points where one may place an item to be added to an existing packing.

The main steps of the algorithm are as follows:

- Order the items according to a sorting criterion
- For each item in the resulting sequence and each rotation r, find the best *EP* of the best available bin where to load the item
- If such a bin exists, load the item into it on the given *EP*
- If the item cannot be loaded in any existing bin, a new bin is created if the total number of bins does not exceed the given maximum, otherwise the item is discarded

Changing the maximum number of available bins adapts EP-BPH to different packing problems. For example, the number of bins is infinite in the Bin Packing problem, but it is equal to 1 in the 3BKP problem. EP-BPH behavior depends on how the best *EP* is selected and how the items are sorted. Computational experiments have shown that, from the *EP* selection point of view, the best trade off between solution quality and computational results is given by the *Residual Space* rule (see Crainic et al. [7]).

The *Residual Space* (*RS*) measures the free space available around an *EP*. Roughly speaking, the *RS* of an *EP* is the distance, along each axis, from the bin edge or the nearest item. The nearest item can be different on each axis. More precisely, when an *EP* is created, its *Residual Space* on each axis is set equal to the distance from its position to the side of the bin along that axis (Figure 7a). The algorithm puts an item on the *EP* that minimizes the difference between its *RS* and the item size:

$$f = [(RS_e^x - w_j) + (RS_e^y - d_j) + (RS_e^z - h_j)], \tag{21}$$

where RS_e^x, RS_e^y, and RS_e^z are the *RS*s of a given *EP* e on x, y, and z axes, respectively. Every time an item is added to the packing, the *RS*s of all the *EP*s are updated. Figure 7b illustrates the concept. For "complex" packings, the *RS* gives only an estimate of the effective volume available around the *EP*s and, thus, potential overlaps with other items have to be checked when accommodating a new item on the chosen *EP*. See Crainic et al. [7] for further details.

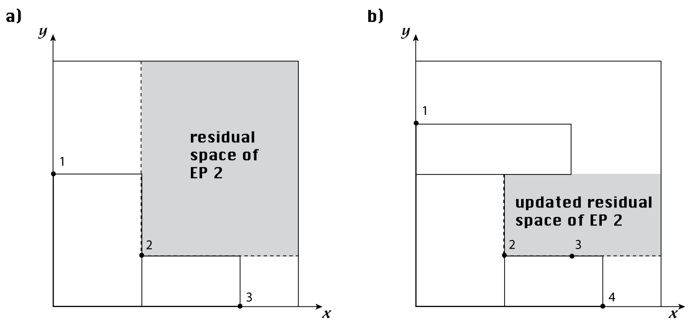

Figure 7: Example of *Residual Space*

To build an initial solution, we apply EP-BPH using a number of sorting criteria. The resulting PCH heuristic builds an initial solution by iteratively applying the sorting criteria and then selecting the best one.

Items may have several attributes, but from the sorting algorithm perspective, the most important ones are:

1. *profit*: the worth or priority of an item;
2. *specific weight, sw*;
3. *area*: for three-dimensional problems it must be meant as the item projection on the (x, y) plane (see Figure 8).

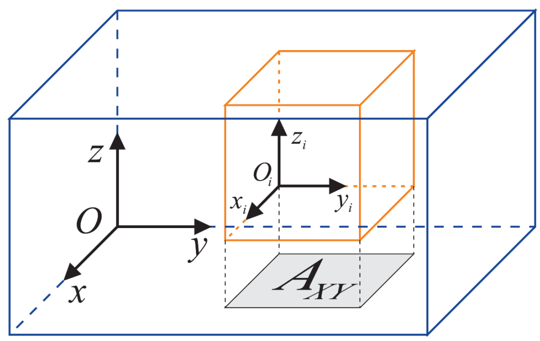

Figure 8: Definition of the item area

Since items show more than one attribute, many ways to sort them are possible. Giving more importance to an attribute means to favor those items showing the highest values of that attribute or score. Often sorting procedure involves more than an attribute or more than a score. Sometimes items sorted afterwards are grouped into *clusters*. A cluster is a set of items showing "close" values of a particular attribute or score. By "close" we mean that the values are inside a given set. Suppose, for instance, to sort the items by clustered area (see Figure 8). Let $A_{\min}$, $A_{\max}$ be the extreme values of the area interval that we want to cluster. Each cluster will have a length which is the length of the global interval $A_{\max} - A_{\min}$ times a given percentage $\theta/100$, with $\theta \in [1, 100]$. The number of clusters n_c is the ratio between the overall interval length and the length of a single cluster. This ratio is $n_c = \lceil 100/\theta \rceil$. Each cluster $A_i(\theta)$ can then be expressed as:

$$A_i(\theta) = [A_{\min} + (i-1)(A_{\max} - A_{\min})\theta/100, \\ A_{\min} + i(A_{\max} - A_{\min})\theta/100], \tag{22}$$

with $i = 1, \ldots, n_c$. Note that, if we want to cluster the overall bin (basis) area, then $A_{\min} = 0$ and $A_{\max} = W \times D$ and (22) becomes:

$$A_i(\theta) = [(i-1)WD\theta/100, \quad iWD\theta/100], \tag{23}$$

with $i = 1, \ldots, n_c$. By combining the three item attributes, six different sorting criteria can be performed:

1. *a-sw*: clustered area, sorted specific weight;
2. *a-p*: clustered area, sorted profit;
3. *sw-a*: clustered specific weight, sorted area;
4. *sw-p*: clustered specific weight, sorted profit;
5. *p-sw*: clustered profit, sorted specific weight;
6. *p-a*: clustered profit, sorted area.

When a solution has been calculated, its corresponding objective function value is given by the following merit function:

$$F = P - \alpha U \tag{24}$$

where P is the total profit of the selected items, U is the unbalancing index given by (4), and α is a nonnegative parameter. Note that (24) is equivalent to a linear combination of two issues: the total profit P and the unbalancing index U. This means that, according to α, attention is also devoted to the balancing constraints, even before the center of mass optimization procedure. For setting the α values see subsection 5.3.

5.2. Center of mass optimization

Given a three-dimensional convex domain inside the bin, the balancing procedure tries to adjust the packed items position so that the packing center of mass lies inside the domain. The heuristic just moves already packed items, therefore neither items are added or removed from the knapsack, nor the overall profit is modified by the procedure. The center of mass optimization heuristic works as follows: first it calculates the position $\vec{r}_{CM}$ of packed items center of mass as reported in equation (2), then it moves one item after another so that $\vec{r}_{CM}$ will move towards the desired position. Two issues arise: where to move an item and how to avoid it overlapping other items and the knapsack edges.

We want to move item i from its actual position $\vec{r}_i = (r^1, r^2, r^3)$ to an unknown new position $\vec{x}_i^{\,\prime} = (r'^1\ r'^2, r'^3)$ such that the overall center of mass moves from its actual position $\vec{r}_{CM}$ to the new desired position $\vec{r}_{CM}^{\,\prime}$ in order to meet the balancing conditions. By (2) the current center of mass can be written as:

$$\vec{r}_{CM} = \sum_{\substack{j \in J' \\ j \neq i}} m_j \vec{r}_{CM_i}/M + m_i \vec{r}_{CM_i}/M \tag{25}$$

When item i, rotated with rotation $\tilde{r}$, moves from $\vec{r}_i$ to $\vec{r}_i^{\,\prime}$ then its new center of mass becomes $\vec{r}_{CM_i}^{\,\prime}$, while the overall center of mass is:

$$\vec{r}_{CM}^{\,\prime} = \sum_{\substack{j \in J' \\ j \neq i}} m_j \vec{r}_{CM_i} / M + m_i \vec{r}_{CM_i}^{\,\prime} / M. \tag{26}$$

Subtracting (25) from (26) we have:

$$\vec{r}_{CM}^{\,\prime} - \vec{r}_{CM} = m_i(\vec{r}_{CM_i}^{\,\prime} - \vec{r}_{CM_i})/M, \tag{27}$$

which leads to the new coordinates of item i center of mass:

$$\vec{r}_{CM_i}^{\,\prime} = \vec{r}_{CM_i} + (\vec{r}_{CM}^{\,\prime} - \vec{r}_{CM})M/m_i. \tag{28}$$

Finally, the new coordinates of item i can be found as:

$$\vec{r}_i^{\,\prime} = (r_{CM_i}^{\prime 1} - \gamma_{i\tilde{r}}^1,\ r_{CM_i}^{\prime 2} - \gamma_{i\tilde{r}}^2,\ r_{CM_i}^{\prime 3} - \gamma_{i\tilde{r}}^3) \tag{29}$$

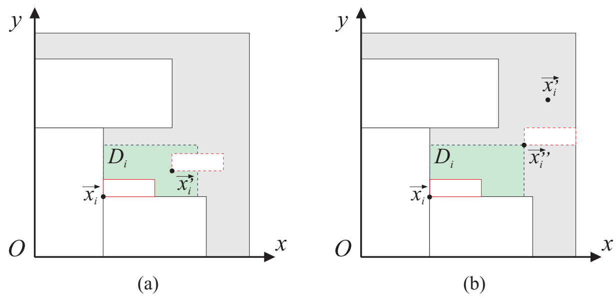

Figure 9: Example of Permitted Movements of an Item in 2D

Unfortunately, due to overlapping issues, it is not always possible to move item i to the position $\vec{r}_i^{\,\prime}$. To overcome this problem a *three-dimensional convex connected domain* is defined where item i can freely move without overlapping neither other items nor the bin. Actually, we define such a domain D_i as the set of allowed positions for the the origin of item i. To do so, an algorithm similar to the one used to calculate the *RS* of an *EP* in Section 5.1 is used. Once D_i has been defined, three possible scenarios may take place:

- $D_i = \{\varnothing\}$: item i cannot move
- $\vec{r}_i^{\,\prime} \in D_i$: item i moves to $\vec{r}_i^{\,\prime}$ thus letting the balancing to be achieved (see Figure 9a for a two-dimensional example)
- $\vec{r}_i^{\,\prime} \notin D_i$ and $D_i \neq \{\varnothing\}$: item i moves to an intermediate position $\vec{r}_i^{\,\prime\prime}$ defined as the point which better approximates $\vec{r}_i^{\,\prime}$ on each axis (see Figure 9b for a two-dimensional example)

Items movements may lead to a state that does not take gravity effects into account. That would result in faulty solutions for many real-life applications, so the algorithm simulates the force of gravity by compacting all items along the z axis towards the (x, y) plan.

The heuristic stops when one of the following three conditions does hold: the packing is balanced, no item can be moved anymore, a maximum number of iterations has been reached.

5.3. Score and parameter setting

In this subsection we show how to set the scores and parameters of the heuristic 3BKP-H.

5.3.1. Score Initialization

The idea is to use the score as a measure of the willingness to accommodate an item into the bin. Consequently, we start from the initial solution decision, and prioritize the items selected by the accommodation procedure by assigning them a higher score than the non loaded ones. Two criteria are used to define such initial scores. First, the score should reflect the profit associated to each item. Second, the gap between a loaded and a non loaded item should be small enough to guarantee the possibility of changes in the ordered list. The initial score of an item is then set to $s_i = kp_i$ if the item has been loaded in the initial solution, and to $s_i = p_i$ otherwise. The value of k has been experimentally set to 3.

5.3.2. Score Update

Previous experience has shown that the various sorting criteria used by the procedure for building the initial solution load into the bin a significant subset of the items making up the optimal solution. "Mistakes" usually are caused when selecting among items with similar profits, but with peculiar sizes, resulting in an underutilization of the bin. The score update focuses on a special subset of items: the less profitable items already loaded and the most profitable non loaded ones. This procedure is similar to the one proposed by Boschetti and Mingozzi [45]. The goal is to force at each iteration swaps between less profitable loaded and profitable non loaded items by changing the scores as follows:

- Find the item k loaded during the last iteration, minimizing $\mu_i = (1 + f_i^l)p_i/(w_i d_i h_i)$, where f_i^l represents the number of iterations item i has been loaded into the bin;
- Update the score of item k to $s_k = (1 - \alpha)s_k$, with $\alpha \in (0, 1)$;
- Find the item l non loaded during the last iteration, maximizing $\theta_i = p_i/(w_i d_i h_i(1 + f_i^u))$, where f_i^u represents the number of iterations item i has not been loaded into the bin;
- Update the score of item l to $s_l = (1 + \beta)s_l, \beta \in (0, 1)$;
- Swap the scores of items k and l;
- Keep the score unchanged for all items $i, i \neq k$ and $i \neq l$;

where, μ_i and θ_i measure the willingness to accommodate an item into the bin, f_i^l and f_i^u maintain a long-term memory of the selected items to avoid always selecting from the same subset of items, and α and β represent the percentage score decrease and increase, respectively, and are experimentally set to 0.1. This procedure ensures that at least two items are swapped at each iteration.

5.3.3. Long-term Score Reinitialization

Given the sorted list of items which built the best solution found so far, we first give a score to each item according to the same rule used in Score Initialization. A fixed number of item pairs (i.e. swaps between two items) are then randomly selected and their scores are swapped.

If the best solution found so far is unfeasible, $\alpha = 2\alpha$ and $\beta = 2\beta$. If the best solution is feasible, $\alpha = \alpha/2$ and $\beta = \beta/2$, if the center of mass lies into the central half of its feasibility domain, while they are unchanged otherwise.

5.3.4. Parameter Initialization and Stopping Criteria

- $\alpha = \beta = 0.1$;
- Long-term Score Reinitialization every 1000 iterations;
- number of item pairs: 5% of the items.

The overall process stops after 5 seconds.

6. Computational results

In this section, we analyze the behaviour of the model and the heuristics in term of solution quality and computational efficiency. As the 3BKP is introduced in this paper for the first time, we introduce in Subsection 6.1 some benchmark instances. The first two sets, namely Set1 and Set2, are obtained by extending the instances in literature for the 3KP, while the third one, Set3, extends the rules used in the previous sets in order to diversify the instances. All the tests have been performed on a Intel I7 2.8 GhZ Workstation with 4 Gb of Ram. The model has been solved by means of Gurobi 4.0 solver limited to 1 core [46]. Subsection 6.2 is devoted to compare the computational results of the MIP model and the heuristic, while subsection 6.3 shows the behaviour of the developed model and heuristic compared with state-of-the-art algorithms. Being the 3BKP a new problem, we compare the model and the heuristic with the results of heuristics developed specifically for the problem which is more similar to the 3BKP, the Three-Dimensional Knapsack Problem.

6.1. Test Instances

In this section we introduce different instance sets for 3BKP. Following the tests for the 3KP, the instances cover up to items and different types of items, knapsack and weight distributions. The sets, namely Set1 and Set2, are obtained by extending the instances by Egeblad and Pisinger [33]. All instance sets can be downloaded from the web site of OR-Library [47]. In the sets Set1 and Set2, the size of the knapsack as well as the size of the items are the same of [33], while weights are considered as additional item attributes. Thus, the two sets differ for the weight generation, i.e. the weights in Set1 are generated in a smaller interval than in Set2. In order to give a better description of the instances, in the following we report the full list of the parameters used to generate the instances:

- *number of items*: $n \in \{20, 40, 60\}$;
- *item generation strategy*: $t \in \{C, R\}$, where:
  - C alias *clustered*, because the instance consists of only 20 items which are duplicated appropriately;
  - R alias *random*, because the instance consists of independently generated items;
- *bin size*: they are calculated such that $W = D$, $H = 2W$ and the bin volume WDH is equal to a percentage p of the total volume of the items. In particular, $p = 50\%$ for Set1 and $p = 90\%$ for Set2;
- *item attributes*:
  - *size*: $s_i = (w_i, d_i, h_i)$, which must belong to one among the following *geometric classes* (see [33]):
    - Cubes (C). The items are cubic and their sizes are defined as $w_i \in [1, 100]$, $d_i = w_i$, $h_i = w_i$;
    - Diverse (D). The sizes of the items are randomly chosen in the following ranges $w_i \in [1, 50]$, $d_i \in [1, 50]$, $h_i \in [1, 50]$;
    - Long (L). The sizes of the items are randomly chosen in the following ranges $w_i \in [1, 200/3]$, $d_i \in [50, 100]$, $h_i \in [1, 200/3]$;
    - Uniform (U). The sizes of the items are randomly chosen in the following ranges $w_i \in [50, 100]$, $d_i \in [50, 100]$, $h_i \in [50, 100]$.
  - *profit*: $p_i = 200 + w_i\, d_i\, h_i$;
  - Center of mass position: the center of mass of each item is placed in the geometrical center of the item itself, i.e. $\vec{r}_{CM_i} = \{w_i/2, d_i/2, h_i/2\}$
  - *specific weight*: sw_i, uniformly distributed in the interval I_{sw}, where the limits of the interval depend on the set:
    - Set1: $I_{sw} = [70, 100]$;
    - Set2: $I_{sw} = [10, 1000]$;
  - *CoM domain*: the domain constraints are set as $L^\delta = \{W/4, D/4, 0\}$ and $U^\delta = \{3W/4, 3D/4, H/2\}$. These limits are given by practical issues in maritime and air cargo applications. In particular, for the limits on z, for stability reasons the requirement is usually as near as possible to 0, i.e. the bottom of the bin [1].

The combination of all the values give 120 instances, 60 for each set.

6.2. *Model and Heuristic results*

This section is devoted to compare the results of the different solution methods for 3BKP, 3BKP-M solved by means of a commercial solver and 3BKP-H.

As stated in Section 4, model 3BKP-M is not efficient in proving the optimality of the solutions due to the poor quality of its continuous relaxation. In our tests we used three well-know commercial solver, FICO XPress 2010, CPLEX 12.1 and Gurobi 4.0. The three solvers have been tested by using their default parameter values, setting a maximum computation time of 200 seconds and a mono-processor setting. Moreover, we tested them with their internal cut generation on and off. We do not report the detailed results of our tests, but we can say that for all solvers the best results in 200 seconds are found by setting off the internal cut generator. Moreover, the solver able to find the best solutions is Gurobi.

According to our tests, Gurobi is quite efficient in finding very good feasible solutions of the MIP model. Moreover, these solutions are found at the very first nodes of the search. On the contrary, if we consider the quality of the model relaxation, Gurobi presents comparable results with respect to XPress and CPLEX.

Table 1 compares the behaviour of UB_{1D} with the continuous relaxation of 3BKP-M. The first column reports the number of items in the instances, while the two remaining columns give the percentage gap of UB_{1D} with respect to the continuous relaxation of 3BKP-M applied to Set1 and Set2 (a positive value means that the continuous relaxation is tighter than UB_{1D}). Each cell reports the mean over the instances with the same number of items. This aggregation is justified by the fact that the number of items has been the only parameter affecting the results. The data show that the gap is large on the 20 item instances only, while it decreases rapidly with the increasing of the items. This is mainly due to constraints (9) and (10), that cluster a lot the binary variables when a continuous relaxation is applied.

Finally, Table 2 is devoted to compare the results obtained by the MIP model 3BKP-M and 3BKP-H, where the computational times have been set to 200 seconds for 3BKP-M and 5 seconds for 3BKP-H. The meaning of the columns is the following:

- Columns 1-4. The columns give the instance name defined in [33], the number of items, the item geometry class and the item generation strategy.
- Columns 5-6. The percentage gap between the solution obtained by 3BKP-H and the continuous relation of 3BKP-M in Set1 and Set2, respectively. When we proved the optimality of the solution by means of 3BKP-M, an asterisk is placed in the table.

The results show how the model performs better than the heuristic on small sized instances (20 items), while starting from 40 items the heuristic is able to give results which are about 10% better than the model and the computational effort of 3BKP-H is about two order of magnitude less than 3BKP-M. Moreover, the accuracy gap between the model and the 3BKP-H in 20 item instances can be reduced by increasing the computational time of 3BKP-H to 10 seconds.

Finally, Table 3 reports the position of the packing center of mass. The values are grouped by item geometry class. Due to the presence of different knapsack sizes and in UB_{1D} order to uniform the results, for each class we give the position along the three axes of the center of mass as percentage with respect to the full size of the knapsack. Thus, the geometric center of the knapsack corresponds to the values 50%, 50%, 50%. According to these results, we can notice that the packing is very well balanced, with its center of mass almost in the center of the feasibility domain along x and y axes, while is in the given domain in the z axis too. This axis is quite peculiar, in fact in practical applications is very difficult to obtain a center of mass in the lower half bottom of the knapsack without losing most of the bin loading volume. In our tests, we had a mean filling ration of 76% in Set2, the most unbalanced, which raises to about 83% if we solve the instances like a standard bin loading problem, i.e. the profit of the items is the volume of the items themselves. These results are very promising, considering that in the unbalanced applications the filling ration is around 92%. Moreover, these gap is mainly due to the constraint on z. In fact, by relaxing it we can fill the bin at 87% in average.

6.3. *State of the Art results*

As stated in Section 3, 3BKP is introduced in this paper for the first time. Thus, no other method than our model and heuristic is present in the literature. Moreover, computing specific upper bounds for 3BKP is quite difficult. In fact, upper bounds obtained by model 3BKP-M are quite poor and have mainly the same quality a trivial bound

n	**Set1**	**Set2**
20	6.16	6.35
40	0.35	1.12
60	0.00	0.00

Table 1: Comparison of different upper bounds

UB_{1D} obtained by computing the optimal solution of the mono-dimensional Knapsack Problem ([37, 33]). Moreover, additional upper bounds that can be obtained by means of conservative scales in the 3D Packing without rotation are not valid for the problems where the rotations are allowed [33]. On the other hand, 3BKP is an extension of the 3KP and thus the solutions obtained by 3BKP-M and 3BKP-H are valid for 3KP. Thus, in Table 4 we compare 3BKP-M and 3BKP-H with the results obtained by H_{EP}, heuristic by Egeblad and Pisinger [33] on their instances for 3KP. The computational times have been set to 120 seconds for H_{EP}, 200 seconds for 3BKP-M and 5 seconds for 3BKP-H and they will be not reported in the table. 3BKP-M is solved by means of Gurobi 4.0 [46], while 3BKP-H is implemented in C++. For H_{EP} the results have been given by [33].

The meaning of the columns is the following:

- Columns 1-4. The instance name defined in [33], the number of items, the item geometry class, and the item generation strategy.
- Column 5. The objective function of the upper bound UB_{1D}, which is the only one computed for all the methods.
- Columns 6-9. The objective function of the best solution found by H_{EP}, the Model 3BKP-M, our heuristic with (3BKP-H) and without (3BKP-H UNB) the balancing constraints activated.
- Columns 10-14. The percentage gap between the upper bound UB_{1D} and objective function of the best solution found by H_{EP}, the Model 3BKP-M, our heuristic with (3BKP-H) and without (3BKP-H UNB) the balancing constraints activated. In the case of 3BKP-H with balancing constraints, we consider the weights of Set1. If we prove the optimality of the solution by means of 3BKP-M, an asterisk is placed in the table.

From the results we can notice that the model is not competitive, with a gap almost doubled than H_{EP}. However, the model is much more flexible than the heuristic, making possible to easily introduce additional constraints like fixed positions for the items, forbidden rotations and precedence constraints in items loading. In fact, while in the heuristic we need additional code to consider these constraints, in the case of the model we simply need to fix the appropriate variables. Moreover, giving to the model a time limit equal to 1000 seconds, the gap can be reduced, even if it is still about 10% higher than H_{EP}. If we compare H_{EP} with 3BKP-H with the balancing constraints activated, we can notice that the results of 3BKP-H are about 3% worse than H_{EP}. However, this gap is given by the balancing constraints. In fact, if we remove the balancing constraints we obtain a total mean gap of 16%, which is about 2% less than Egeblad and Pisinger results. These results are more impressive if we consider that 3BKP-H require a computational time which is about 2 order of magnitude less than H_{EP}. We also tried to increase the computational time of 3BKP-H in order to obtain better results, but the computational experience show that the increase of quality is negligible. From the means computed per item number, we can see how the gap between H_{EP} and 3BKP-H is constantly present in all the instances, even if it reduces while the size of the instances increases. 3BKP-M is competitive when the number of items is 20, but its gaps makes it unusable in practice for larger instances. Finally, the consistent gap between UB_{1D} and all the presented methods is, how stated in [33], mainly due to the poor quality of the relaxation, which is not able to take into account neither geometric nor balancing issues.

7. Conclusions

In this paper, we introduced the Three-Dimensional Knapsack Problem with Balancing Constraints, the extension of the Three-Dimensional Knapsack Problem (3KP) where additional constraints related to the Center of Mass of the three-dimensional packing are given. A MIP formulation of the problem as well as an efficient and accurate heuristic

Instance	n	IT Geom	IT Strat	3BKP-M		3BKP-H	
				Set1	Set2	Set1	Set2
ep3d-20-C-C-50	20	C	C	7.0	7.0	7.0	7.0
ep3d-20-C-C-90	20	C	C	*	*	*	*
ep3d-20-C-R-50	20	C	R	*	*	*	*
ep3d-20-C-R-90	20	C	R	6.3	10.3	8.2	12.0
ep3d-20-D-C-50	20	D	C	*	*	0.2	4.0
ep3d-20-D-C-90	20	D	C	17.8	19.6	26.7	27.4
ep3d-20-D-R-50	20	D	R	14.6	22.9	11.0	11.0
ep3d-20-D-R-90	20	D	R	16.8	14.7	19.1	19.1
ep3d-20-F-C-50	20	F	C	20.1	18.9	20.7	20.7
ep3d-20-F-C-90	20	F	C	26.4	23.9	27.6	27.6
ep3d-20-F-R-50	20	F	R	22.9	17.9	20.1	20.1
ep3d-20-F-R-90	20	F	R	20.6	21.0	21.2	21.2
ep3d-20-L-C-50	20	L	C	11.6	13.2	16.3	19.1
ep3d-20-L-C-90	20	L	C	13.2	12.1	14.5	14.5
ep3d-20-L-R-50	20	L	R	11.9	10.0	12.8	12.8
ep3d-20-L-R-90	20	L	R	13.9	15.1	17.4	19.3
ep3d-20-U-C-50	20	U	C	23.8	23.8	30.4	30.4
ep3d-20-U-C-90	20	U	C	21.0	14.7	24.6	26.8
ep3d-20-U-R-50	20	U	R	21.0	16.7	30.0	29.5
ep3d-20-U-R-90	20	U	R	21.0	18.5	17.4	17.4
ep3d-40-C-C-50	40	C	C	38.7	38.7	38.7	38.7
ep3d-40-C-C-90	40	C	C	21.6	*	24.6	3.0
ep3d-40-C-R-50	40	C	R	24.3	24.7	26.7	33.1
ep3d-40-C-R-90	40	C	R	24.1	26.1	30.1	32.7
ep3d-40-D-C-50	40	D	C	27.6	26.6	19.9	18.2
ep3d-40-D-C-90	40	D	C	37.6	24.4	21.0	21.0
ep3d-40-D-R-50	40	D	R	35.3	37.3	12.7	11.5
ep3d-40-D-R-90	40	D	R	32.1	38.7	15.9	15.9
ep3d-40-F-C-50	40	F	C	31.5	29.0	32.0	31.1
ep3d-40-F-C-90	40	F	C	37.6	33.4	33.5	33.5
ep3d-40-F-R-50	40	F	R	29.9	26.4	19.3	19.3
ep3d-40-F-R-90	40	F	R	38.8	26.1	22.1	22.1
ep3d-40-L-C-50	40	L	C	33.2	33.5	17.2	15.9
ep3d-40-L-C-90	40	L	C	35.4	41.5	20.6	21.4
ep3d-40-L-R-50	40	L	R	37.9	32.9	12.1	9.9
ep3d-40-L-R-90	40	L	R	44.4	39.1	16.6	16.6
ep3d-40-U-C-50	40	U	C	21.9	27.6	18.2	17.9
ep3d-40-U-C-90	40	U	C	23.6	20.9	33.4	33.4
ep3d-40-U-R-50	40	U	R	20.4	25.5	13.0	13.0
ep3d-40-U-R-90	40	U	R	31.7	27.5	28.4	28.4
ep3d-60-C-C-50	60	C	C	50.9	53.1	50.9	50.9
ep3d-60-C-C-90	60	C	C	35.9	31.1	29.5	29.5
ep3d-60-C-R-50	60	C	R	29.1	25.3	31.7	31.7
ep3d-60-C-R-90	60	C	R	44.7	36.2	25.2	25.2
ep3d-60-D-C-50	60	D	C	23.8	29.4	20.5	20.5
ep3d-60-D-C-90	60	D	C	29.7	28.1	30.6	30.6
ep3d-60-D-R-50	60	D	R	50.0	38.7	6.7	6.7
ep3d-60-D-R-90	60	D	R	49.1	53.3	12.2	12.2
ep3d-60-F-C-50	60	F	C	33.5	41.6	22.6	22.7
ep3d-60-F-C-90	60	F	C	29.9	36.8	35.7	34.3
ep3d-60-F-R-50	60	F	R	32.6	30.8	19.1	20.1
ep3d-60-F-R-90	60	F	R	40.9	30.0	27.8	27.8
ep3d-60-L-C-50	60	L	C	33.1	36.1	17.3	17.3
ep3d-60-L-C-90	60	L	C	38.9	39.1	22.9	21.0
ep3d-60-L-R-50	60	L	R	63.3	45.2	9.8	9.8
ep3d-60-L-R-90	60	L	R	51.0	41.5	17.1	17.1
ep3d-60-U-C-50	60	U	C	24.1	26.9	20.2	20.2
ep3d-60-U-C-90	60	U	C	37.6	30.4	39.7	39.4
ep3d-60-U-R-50	60	U	R	31.0	35.3	17.7	17.7
ep3d-60-U-R-90	60	U	R	37.3	31.4	35.1	35.2

Table 2: Comparison of 3BKP-M and 3BKP-H

IT Geom	CM_X	CM_Y	CM_Z
C	56.82%	53.95%	49.72%
D	53.95%	52.23%	44.82%
F	53.99%	51.91%	45.56%
L	53.40%	52.24%	44.60%
U	55.04%	53.05%	45.82%

Table 3: Center of Mass position

have been presented. Extensive computational results showed how the MIP model is able to find better bounds than other relaxations and the heuristic is able to efficiently solve both instances explicitly designed for the 3BKP, as well as to be competitive with methods explicitly designed to solve the 3KP. Presently, we are extending the test instances in order to give a better insight of the relationship between solution quality and balancing constraints tightness.

Acknowledgments

We express our gratitude to Prof. David Pisinger, who graciously provided us with sets of test instances, as well as their detailed results. Moreover, the authors are grateful to Roberto Marcellino for his contribution to a previous version of the paper. This project has been partially supported by the Ministero dell'Istruzione, Università e Ricerca (MIUR) (Italian Ministry of University and Research), under the Progetto di Ricerca di Interesse Nazionale (PRIN) 2007 "Optimization of Distribution Logistics", and the Natural Sciences and Engineering Council of Canada (NSERC), through its Industrial Research Chair and Discovery Grants programs, and by the partners of the Chair, CN, Rona, Alimentation Couche-Tard and the Ministry of Transportation of Québec.

References

[1] B. L. Kaluzny, R. H. A. D. Shaw, Optimal aircraft load balancing, International Transactions in Operational Research 16 (2009) 767–787.
[2] L. Colaneri, F. Della Croce, G. Perboli, R. Tadei, A heuristic procedure for the automated cargo vehicle loading problem: A case study, in: T. A. Ciriani (Ed.), OR in Space and Air, OR in Space and Air, Springer, 2003, pp. 27–42.
[3] E. E. Bischoff, M. S. W. Ratcliff, Issues in the development of approaches to container loading, Omega 23 (1995) 377–390.
[4] M. Mongeau, C. Bés, Optimization of aircraft container loading, IEEE Transactions on Aerospace and Electronic Systems 39 (2003) 140–150.
[5] T. G. Crainic, G. Perboli, R. Tadei, Unipack: a new heuristic framework for multi-dimensional packing problems, in: Proceedings of MIC 2007 The Seventh Metaheuristics International Conference, MIC 2007, 2007, pp. 1–6.
[6] G. Perboli, T. G. Crainic, R. Tadei, An efficient metaheuristic for multi-dimensional multi-container packing, in: IEEE (Ed.), IEEE Conference on Automation Science and Engineering (CASE) 2011, 2011, pp. 563–568.
[7] T. G. Crainic, G. Perboli, R. Tadei, Extreme point-based heuristics for three-dimensional bin packing, INFORMS Journal on Computing 20 (2008) 368–384.
[8] L. Junqueira, R. Morabito, D. S. Yamashita, Three-dimensional container loading models with cargo stability and load bearing constraints, Computers & Operations Research 39 (2012) 74–85.
[9] G. Wäscher, H. Hauβner, H. Schumann, An improved typology of cutting and packing problems, European Journal of Operational Research 183 (2007) 1109–1130.
[10] H. Onodera, Y. Taniguchi, K. Tamaru, Branch-and-bound placement for building block layout, Proc. 28th ACM/IEEE Design Automation Conf., ACM Press, New York (1991) 433–439.
[11] C. S. Chen, S. M. Lee, Q. S. Shen, An analytical model for the container loading problem, European Journal of Operational Research 80 (1995) 68–76.
[12] D. Pisinger, M. Sigurd, Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem, INFORMS Journal on Computing 19 (2007) 36–51.
[13] H. Dyckhoff, A typology of cutting and packing problems, European Journal of Operational Research 44 (1990) 145–159.
[14] P. C. Gilmore, R. E. Gomory, Multistage cutting stock problems of two and more dimensions, Operations Research 13 (1965) 94–120.
[15] R. Baldacci, M. A. Boschetti, A cutting plane approach for the two-dimensional orthogonal non-guillotine cutting problem, European Journal of Operational Research 183 (2007) 1136–1149.
[16] J. E. Beasley, An exact two-dimensional non-guillotine cutting stock tree search procedure, Operations Research 33 (1985) 49–64.
[17] F. K. R. Chung, M. R. Garey, D. S. Johnson, On packing two-dimensional bins, SIAM - Journal of Algebraic and Discrete Methods 3 (1982) 66–76.
[18] J. O. Berkey, P. Y. Wang, Two dimensional finite bin packing algorithms, Journal of the Operational Research Society 38 (1987) 423–429.
[19] S. P. Fekete, J. Schepers, New classes of lower bounds for bin packing problems, Math. Programming 91 (1) (2001) 11–31.
[20] S. Martello, D. Pisinger, D. Vigo, The three-dimensional bin packing problem, Operations Research 48 (2) (2000) 256–267.

Instance	n	IT Geom	IT Strat	UB_{1D}	H_{EP}	3BKP-M	3BKP-H	3BKP-H UNB	H_{EP}	3BKP-M	3BKP-H	3BKP-H UNB
ep3d-20-C-C-50	20	C	C	1026348	633672	633672	633672	633672	7.0	7.0	7.0	7.0
ep3d-20-C-C-90	20	C	C	1834340	916241	916241	916241	916241	*	*	*	*
ep3d-20-C-R-50	20	C	R	2188245	1492413	1492413	1492413	1492413	*	*	*	*
ep3d-20-C-R-90	20	C	R	3925057	2497691	2497691	2449089	2497691	*	*	1.9	*
ep3d-20-D-C-50	20	D	C	395916	239532	316492	315964	315964	24.3	*	0.2	0.2
ep3d-20-D-C-90	20	D	C	718692	468112	559756	526480	526480	34.9	22.1	26.7	26.7
ep3d-20-D-R-50	20	D	R	240621	195937	206026	214227	214227	18.6	14.4	11.0	11.0
ep3d-20-D-R-90	20	D	R	414188	318848	368476	335123	335123	23.0	11.0	19.1	19.1
ep3d-20-F-C-50	20	F	C	2395087	1900250	1900250	1900250	1900250	20.7	20.7	20.7	20.7
ep3d-20-F-C-90	20	F	C	4304020	2989393	2510887	3118132	3118132	30.5	41.7	27.6	27.6
ep3d-20-F-R-50	20	F	R	2252037	1563997	1698795	1800399	1800399	30.5	24.6	20.1	20.1
ep3d-20-F-R-90	20	F	R	4099982	2918002	2353072	3232580	3232580	28.8	42.6	21.2	21.2
ep3d-20-L-C-50	20	L	C	1064487	834335	941069	891283	891283	21.6	11.6	16.3	16.3
ep3d-20-L-C-90	20	L	C	1894489	1589303	1546821	1619190	1619190	16.1	18.4	14.5	14.5
ep3d-20-L-R-50	20	L	R	718561	569900	633463	626381	626381	20.7	11.8	12.8	12.8
ep3d-20-L-R-90	20	L	R	1282710	1051084	1114602	1056699	1056699	17.9	12.9	17.5	17.5
ep3d-20-U-C-50	20	U	C	4495440	3088676	2796072	3127252	3127252	31.3	37.8	30.4	30.4
ep3d-20-U-C-90	20	U	C	8067424	5360280	4289980	6074000	6074000	33.5	46.7	24.6	24.6
ep3d-20-U-R-50	20	U	R	4413077	3509748	2677316	3087569	3509748	20.5	39.3	30.0	20.5
ep3d-20-U-R-90	20	U	R	8041072	6921250	4156121	6638762	6921250	13.9	48.3	17.4	13.9
Mean									**19.7**	**20.5**	**15.9**	**15.2**
ep3d-40-C-C-50	40	C	C	2065540	1265664	1265664	1265664	1265664	38.7	38.7	38.7	38.7
ep3d-40-C-C-90	40	C	C	3652448	2828160	2335561	2717385	2828160	21.8	35.4	24.8	21.8
ep3d-40-C-R-50	40	C	R	4102972	3002269	2503936	3008658	3008658	26.8	39.0	26.7	26.7
ep3d-40-C-R-90	40	C	R	7335602	5972946	3498247	4900577	5972946	4.1	43.8	21.3	4.1
ep3d-40-D-C-50	40	D	C	788124	539040	580512	630996	630996	31.6	26.3	19.9	19.9
ep3d-40-D-C-90	40	D	C	1423896	1126300	1032016	1124788	1126300	20.9	27.5	21.0	20.9
ep3d-40-D-R-50	40	D	R	399894	349470	248518	349152	349470	12.6	37.9	12.7	12.6
ep3d-40-D-R-90	40	D	R	728248	639819	510962	612487	639819	12.1	29.8	15.9	12.1
ep3d-40-F-C-50	40	F	C	4816926	3590244	2596874	3274502	3590244	25.5	46.1	32.0	25.5
ep3d-40-F-C-90	40	F	C	8664122	6435962	4039655	5760960	6435962	25.7	53.4	33.5	25.7
ep3d-40-F-R-50	40	F	R	4518343	3477469	2623783	3644680	3644680	23.0	41.9	19.3	19.3
ep3d-40-F-R-90	40	F	R	8199224	7336067	3560051	6386094	7336067	10.5	56.6	22.1	10.5
ep3d-40-L-C-50	40	L	C	2127316	1675122	1410197	1760700	1760700	21.3	33.7	17.2	17.2
ep3d-40-L-C-90	40	L	C	3819412	2943657	2054950	3032364	3032364	22.9	46.2	20.6	20.6
ep3d-40-L-R-50	40	L	R	1784686	1609648	1067546	1567893	1609648	9.8	40.2	12.1	9.8
ep3d-40-L-R-90	40	L	R	3224295	2699629	1722617	2689260	2699629	16.3	46.6	16.6	16.3
ep3d-40-U-C-50	40	U	C	8988536	7008136	4317064	7355808	7355808	22.0	52.0	18.2	18.2
ep3d-40-U-C-90	40	U	C	16241380	14065676	5580692	10819676	14065676	13.4	65.6	33.4	13.4
ep3d-40-U-R-50	40	U	R	8666294	7766238	4418573	7538465	7766238	10.4	49.0	13.0	10.4
ep3d-40-U-R-90	40	U	R	15531980	13077284	6217878	11120608	13077284	15.8	60.0	28.4	15.8
Mean									**19.3**	**43.5**	**22.4**	**18.0**
ep3d-60-C-C-50	60	C	C	3063219	1504980	1370916	1504980	1504980	50.9	55.2	50.9	50.9
ep3d-60-C-C-90	60	C	C	5517671	4475024	2590702	3892171	4475024	18.9	53.0	29.5	18.9
ep3d-60-C-R-50	60	C	R	6493464	5695120	2916398	4435949	5695120	12.3	55.1	31.7	12.3
ep3d-60-C-R-90	60	C	R	11675188	10209801	4213641	8729652	10209801	12.5	63.9	25.2	12.5
ep3d-60-D-C-50	60	D	C	1200408	1057032	801200	954856	1057032	11.9	33.3	20.5	11.9
ep3d-60-D-C-90	60	D	C	2143544	1843584	1440492	1488020	1843584	14.0	32.8	30.6	14.0
ep3d-60-D-R-50	60	D	R	538113	484363	323947	502275	502275	10.0	39.8	6.7	6.7
ep3d-60-D-R-90	60	D	R	966582	861655	433736	848299	861655	10.9	55.1	12.2	10.9
ep3d-60-F-C-50	60	F	C	7193700	6257697	3700025	5565875	6257697	13.0	48.6	22.6	13.0
ep3d-60-F-C-90	60	F	C	12913715	10412682	3714761	8298024	10412682	19.4	71.2	35.7	19.4
ep3d-60-F-R-50	60	F	R	6780100	6146420	3193484	5484876	6146420	9.3	52.9	19.1	9.3
ep3d-60-F-R-90	60	F	R	12301636	10866326	4154808	8884312	10866326	11.7	66.2	27.8	11.7
ep3d-60-L-C-50	60	L	C	3211612	2327139	1708786	2656622	2656622	27.5	46.8	17.3	17.3
ep3d-60-L-C-90	60	L	C	5736894	4832080	2773026	4422760	4832080	15.8	51.7	22.9	15.8
ep3d-60-L-R-50	60	L	R	2391507	2042317	1157278	2158105	2158105	14.6	51.6	9.8	9.8
ep3d-60-L-R-90	60	L	R	4304649	3872594	1748761	3568203	3872594	10.0	59.4	17.1	10.0
ep3d-60-U-C-50	60	U	C	13508800	12033592	4939888	10782744	12033592	10.9	63.4	20.2	10.9
ep3d-60-U-C-90	60	U	C	24342664	19787768	3868824	14683564	19787768	18.7	84.1	39.7	18.7
ep3d-60-U-R-50	60	U	R	12097660	10857656	5207450	9952696	10857656	10.2	57.0	17.7	10.2
ep3d-60-U-R-90	60	U	R	21893096	19304585	4374061	14216597	19304585	11.8	80.0	35.1	11.8
Mean									**15.7**	**56.1**	**24.6**	**14.8**
Total mean									**18.2**	**40.0**	**21.0**	**16.0**

Table 4: 3KP: results of 3BKP-M and 3BKP-H without balancing constraints compared to H_{EP}

[21] E. den Boef, J. Korst, S. Martello, D. Pisinger, D. Vigo, Erratum to the three-dimensional bin packing problem: Robot-packable and orthogonal variants of packing problems, Operations Research 53 (4) (2005) 735–736.
[22] S. Martello, D. Pisinger, D. Vigo, E. den Boef, J. Korst, Algorithms for general and robot-packable variants of the three-dimensional bin packing problem, ACM Transactions on Mathematical Software 33 (2007) 7–19.
[23] C. Musès, The dimensional family approach in (hyper)sphere packing: A typological study of new patterns, structures, and interdimensional functions, Applied Mathematics and Computation 88 (1997) 1 – 26.
[24] T. Crainic, G. Perboli, R. Tadei, Recent advances in multi-dimensional packing problems, in: C. Volosencu (Ed.), New Technologies - Trends, Innovations and Research. ISBN: 978-953-51-0480-3, InTech, 2012, pp. 91–110.
[25] E. Hadjiconstantinou, N. Christofides, An exact algorithm for general and orthogonal and two-dimensional knapsack problems, European Journal of Operational Research 83 (1995) 39–56.
[26] M. A. Boschetti, E. Hadjiconstantinou, A. Mingozzi, New upper bounds for the two-dimensional orthogonal cutting stock problem, IMA Journal of Management Mathematics 13 (2002) 95–119.
[27] J. A. George, D. F. Robinson, A heuristic for packing boxes into a container, Computers & Operations Research 7 (1980) 147–156.
[28] S. P. Fekete, J. Schepers, A new exact algorithm for general orthogonal d-dimensional knapsack problems, ESA '97, Springer Lecture Notes in Computer Science 1284 (1997) 144–156.
[29] S. P. Fekete, J. Schepers, A combinatorial characterization of higher-dimensional orthogonal packing, Mathematics of Operations Research 29 (2004) 353–368.
[30] A. Caprara, M. Monaci, On the two-dimensional knapsack problem, Operations Research Letters 32 (2004) 5–14.
[31] F. Clautiaux, J. Carlier, A. Moukrim, A new exact method for the two-dimensional orthogonal packing problem, European Journal of Operational Research 183 (2007) 1196–1211.
[32] G. Mavrotas, J. R. Figueira, K. Florios, Solving the bi-objective multi-dimensional knapsack problem exploiting the concept of core, Applied Mathematics and Computation 215 (2009) 2502 – 2514.
[33] J. Egeblad, D. Pisinger, Heuristic approaches for the two- and three-dimensional knapsack packing problem, Computers & Operations Research 36 (2009) 1026–1049.
[34] R. Daughtrey, T. Fennel, D. Schwaab, A simulated annealing approach to 3-D packing with multiple constraints, Tech. Rep. Cosmic Program MFS-287000, Boeing Huntsville Al Center (1991).
[35] Boeing Defence & Space Group, Cargo planning, analysis and configuration system, Tech. Rep. S 683-30014-1, Boeing (1996).
[36] G. Fasano, Cargo analytical integration in space engineering: A three-dimensional packing model, in: T. Ciriani, S. Gliozzi, E. L. Johnson, R. Tadei (Eds.), Operations Research in Industry, Macmillan, 1999, pp. 232–246.
[37] G. Fasano, A MIP approach for some practical packing problems: Balancing constraints and tetris-like items, 4OR: A Quarterly Journal of Operations Research 2 (2) (2004) 161–174.
[38] H. P. Williams, Model Building in Mathematical Programming, John Wiley & Sons, London, 1993.
[39] L. A. MartinVega, Aircraft load planning and the computer: description and review, Computers and Industrial Engineering 4 (1985) 357–369.
[40] D. Cochard, K. Yost, Improving utilization of air force cargo loading, Interfaces 15 (1985) 53–68.
[41] S. Amiouny, J. Bartholdi, J. V. Vate, J. Zhang, Balanced loading, Operations Research 40 (2) (1992) 238–246.
[42] K. Mathur, An integer-programming-based heuristic for the balanced loading problem, Operations Research Letters 22 (1998) 19–25.
[43] E. K. Burke, G. Kendall, G. Whitwell, A new placement heuristic for the orthogonal stock cutting problem, Operations Research 52 (2004) 655–671.
[44] E. K. Burke, G. Kendall, G. Whitwell, A simulated annealing enhancement of the best-fit heuristic for the orthogonal stock cutting problem, INFORMS Journal on Computing 21 (2009) 505–516.
[45] M. A. Boschetti, A. Mingozzi, The two-dimensional finite bin packing problem. part ii: New lower and upper bounds, 4OR 1 (2003) 135–147.
[46] Gurobi Optimization Inc., Gurobi Optimizer Reference Manual Version 4.0., Gurobi Optimization, Houston, 2011.
[47] J. E. Beasley, Or-library: distributing test problems by electronic mail, Journal of the Operational Research Society 41 (1990) 1069–1072. URL `http://people.brunel.ac.uk/ mastjjb/jeb/info.html`