4OR (2008) 6:291–310
DOI 10.1007/s10288-007-0049-1

INDUSTRY

MIP-based heuristic for non-standard 3D-packing problems

Giorgio Fasano

Received: 21 April 2006 / Revised: 6 March 2007 / Published online: 18 July 2007

Abstract This paper is the continuation of a previous work (Fasano in 4OR 2: 161–174, 2004), dedicated to a MIP formulation for *non-standard* 3D-packing issues, with additional conditions. The *Single Bin Packing* problem (*Basic Problem*) is considered and its MIP formulation shortly surveyed, together with some possible extensions, including balancing, *tetris*-like items and non-standard domains. A MIP-based heuristic is proposed to solve efficiently the *Basic Problem* or any possible extension of it, susceptible to a MIP formulation. The heuristic is a recursive procedure based on a *non-blind* local search philosophy. The concept of *abstract configuration*, concerning the *relative positions* between items, is introduced: the *relative positions* of items, determined by any *abstract configuration*, give rise to a feasible solution in an unbounded domain. The heuristic generates a sequence of *good abstract configurations* and solves, step by step, a reduced MIP model by fixing the *relative positions* of items, corresponding to the current *abstract configuration*.

Keywords *Non-standard* 3D-packing · *Balancing* conditions · Additional conditions · MIP-based heuristics · *Abstract configuration*

MSC classification (2000) 05B40 · 90C11 · 90C59 · 90C90 · 68T20

1 Introduction

This paper extends a previous work (Fasano 2004) concerning *non-standard* 3D-packing problems, in the presence of additional conditions. Both works originate from a research activity performed by Alenia Alcatel Space Italia S.p.A., a leading

G. Fasano (✉)
CMath FIMA CSci, Thales Alenia Space Italia S.p.A., Str. Antica di Collegno 253, 10146 Turin, Italy
e-mail: giorgio.fasano@thalesaleniaspace.com

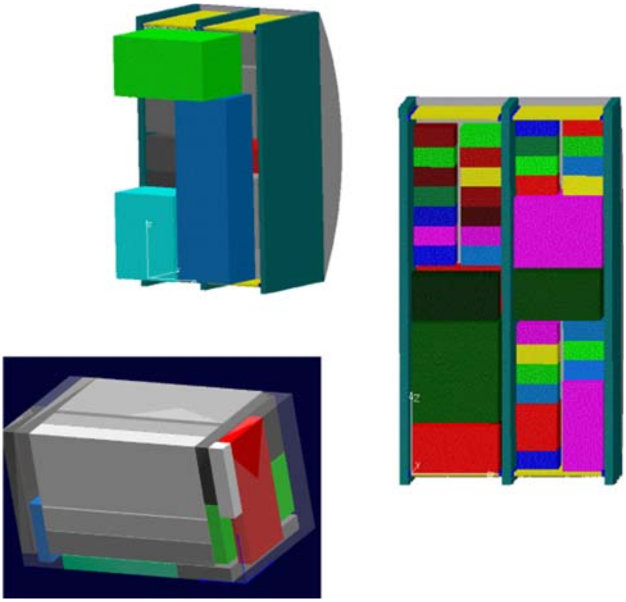


Fig. 1 ATV *cargo accommodation* (*bag/rack* level)

company in European space technology, in support to the *cargo accommodation* of space vehicles and modules. This research activity has been addressed, in particular, in the context of the automated transfer vehicle (ATV) space program (funded by the European Space Agency, ESA) and the CAST project (Fasano et al. 2003).

The ATV is the European transportation system supporting the *International Space Station*. On the basis of the *Cargo Manifest* plan provided by *NASA*, defining the fluids and items quantity to be transported to and from the *Space Station*, a detailed *cargo accommodation* analysis has to be performed, for each launch and for each ATV carrier. The variety of items to consider, as well as the presence of a significant number of complex accommodation rules and requirements, in addition to tight balancing conditions, make the problem very challenging.

The necessity of finding efficient solutions leads frequently to compare different operational scenarios, in order to select the most suitable one.

Last minute upgrades, due to possible re-planning of the *Cargo Manifest*, may moreover arise and this implies the further capability to readapt quickly the *cargo accommodation*. High efficiency and cost effectiveness are thus major concerns, so that looking into valid solutions, by means of a *manual* approach, would represent an impractical job even for expert designers. Responding to these necessities, the European Space Agency funded the development of CAST (Cargo Accommodation Support Tool), an advanced packing tool conceived and developed by Alcatel Alenia Space Italia S.p.A. and devoted to the ATV *cargo accommodation*.

The ATV case involves different *cargo accommodation* levels (see Fig. 1): *small items* have to be accommodated into *bags*; *bags* and *large items* into *racks* or on the *rack fronts*, while *racks* have to be positioned into predisposed locations, inside the

ATV cargo carrier. Mass and volume capacity limitations (at *bag*, *rack* and cargo carrier level) are set, together with specific positioning rules, as well as *static* and *dynamic balancing* conditions. *Small items* and *bags* can be assumed to be parallelepipeds, while, generally, *large items* (for reasons of shape and dimension) have to be treated as clusters of parallelepipeds (mutually orthogonal, such as *tetris*-like items); *racks* are convex domains, subdivided into sectors (Fasano et al. 2003).

Similar problems have to be solved for the *cargo accommodation* of space vehicles and modules, in general. This task poses a number of *non-standard* 3D-packing issues, in the presence of additional conditions, that can arise in several application fields, not limited to space engineering (e.g., aeronautical, naval and transportation systems, logistics, manufacturing, engineering of complex systems).

The literature on the optimization of multidimensional packing problems (well known for being *NP-hard*) is widespread and advanced methods are available to solve efficiently difficult instances (Coffman et al. 1997; Dyckhoff et al. 1997; Fekete and Schepers 2004, 2006; Martello et al. 2000, 2006; Pisinger 1998; Pisinger and Sigurd 2006). Even if works concerning non-standard packing problems are available in the literature (Addis et al. 2006; Blazewicz et al. 1993; Castillo et al. 2006; Egeblad et al. 2006; Gones and Oliveira 2002; Oliveira and Ferreira 1993; Pintér and Kampas 2004, 2005; Stoyan et al. 2004), most of the research focuses on the orthogonal placement of rectangular items into rectangular domains, with no additional constraints. Non-standard packing problems with additional constraints are often tackled by dedicated heuristics or meta-heuristics (Daughtrey et al. 1991; Tadei et al. 2003; Takadama et al. 2004), but possible approaches, based on Mixed Integer Programming (MIP), have also been investigated (Chen et al. 1995; Fasano 1989, 1999, 2003, 2004; Fischetti and Luzzi 2003; Mathur 1998; Onodera et al. 1991; Padberg 1999; Pisinger and Sigurd 2005).

A MIP approach has been introduced (Fasano 1999, 2003, 2004) to formulate non-standard packing problems, involving non-rectangular domains and *tetris*-like items, in the presence of additional constraints (e.g., separation planes, fixed position or orientation of specific items, *static balancing*).

The author's previous work (Fasano 2004) focused on the modeling aspects. In it, it has been shown that the classical *Three-dimensional Single Bin Packing* problem, as well as a quite large class of non-standard packing issues, can be quite easily contemplated by a MIP formulation. The present work focuses on a heuristic approach aimed at efficiently solving such MIP models. It is based on a recursive procedure that solves, at each step, a simplified MIP model, replacing the original one. This procedure reduces, at each step, the difficulties associated to the item-item non-intersection conditions. Being such conditions mandatory in any packing problem, the procedure is applicable to any non-standard packing problem, provided that it is susceptible to a MIP formulation, independently from the non-standard conditions to consider. This paper is thus concentrated on the heuristic logic more than on the modeling aspects related to the non-standard conditions. They are nevertheless explicitly reviewed, referring to the quoted works for a more detailed analysis.

The classical *Three-Dimensional Single Bin Packing* problem is considered first, in Sect. 2.1. It consists of placing (orthogonally and with possibility of rotation) parallelepipeds (from a given set) into a parallelepiped, maximizing the loaded volume (or mass). The modeling aspects relative to non-standard problems are surveyed in Sect. 2.2 and the intrinsic difficulties related to the MIP formulation in Sect. 2.3. Section 3 is dedicated to the MIP-based heuristic. It is aimed at solving efficiently, in terms of computational time and goodness of the solutions, any *non-standard* three-dimensional packing problem, susceptible to a MIP formulation. Section 3.1 outlines the basic concept of the heuristic procedure whose logic is described in Sect. 3.2. Section 3.3 refers to the experimental analysis and application aspects.

2 A MIP approach for *non-standard* 3D-packing problems

2.1 The *Basic Problem*

The classical *Three-dimensional Single Bin Packing* problem, denoted here as *Basic Problem* can be described as follows. Given a set of n parallelepipeds (items with homogeneous density) and a parallelepiped D (domain), place items into D maximizing the loaded volume (or mass), with the following positioning rules (for picked items):

- *each parallelepiped side has to be parallel to a side of D* (*orthogonality* conditions)
- *each parallelepiped has to be contained within D (domain* conditions)
- *parallelepipeds cannot overlap* (*non-intersection* conditions)

The *Basic Problem* (with possible variations) can be easily formulated as a MIP problem (see Chen et al. 1995; Fasano 1999, 2004, 1998; Onodera et al. 1991; Padberg 1999; Pisinger and Sigurd 2005). For each parallelepiped i denote by L_{1i}, L_{2i}, L_{3i}, with $L_{1i} \leq L_{2i} \leq L_{3i}$, its sides and by w_{1i}, w_{2i}, w_{3i} the coordinates of its center (coincident with the center of mass), with respect to a predefined orthonormal reference frame (with origin O and axes w_1, w_2, w_3). The domain D is a parallelepiped with sides D_1, D_2, D_3, parallel to the w_1, w_2, w_3 reference frame axes respectively. A vertex of D is, moreover, supposed to be coincident with the reference frame origin O and D lies within the positive quadrant of the reference frame. By setting $\alpha \in \{1, 2, 3\}$, $\beta \in \{1, 2, 3\}$, $i \in \{1, \ldots, n\}$, the binary variables $\chi_i, \delta_{\alpha\beta i} \in \{0, 1\}$ are introduced with the following meaning:

$\chi_i = 1$ if item i is picked, $\chi_i = 0$ otherwise;

$\delta_{\alpha\beta i} = 1$ if $L_{\alpha i}$ is parallel to the w_β axis, $\delta_{\alpha\beta i} = 0$ otherwise.

The *Basic Model* is formulated as follows (Fasano 2004).

Orthogonality constraints:

$$\forall\alpha, \forall i \quad \sum_{\beta=1}^{3} \delta_{\alpha\beta i} = \chi_i \tag{1}$$

$$\forall\beta, \forall i \quad \sum_{\alpha=1}^{3} \delta_{\alpha\beta i} = \chi_i \tag{2}$$

Domain constraints:

$$\forall\beta, \forall i \quad 0 \le w_{\beta i} - \frac{1}{2}\sum_{\alpha=1}^{3} L_{\alpha i}\delta_{\alpha\beta i} \le w_{\beta i} + \frac{1}{2}\sum_{\alpha=1}^{3} L_{\alpha i}\delta_{\alpha\beta i} \le D_\beta \chi_i. \tag{3}$$

N*on-intersection* constraints:

$$\forall\beta, \forall i, \forall j, i < j \quad w_{\beta i} - w_{\beta j} \ge \frac{1}{2}\sum_{\alpha=1}^{3} \left(L_{\alpha i}\delta_{\alpha\beta i} + L_{\alpha j}\delta_{\alpha\beta j}\right) - \left(1 - \sigma^{+}_{\beta ij}\right) D_\beta \tag{4-1}$$

$$\forall\beta, \forall i, \forall j, i < j \quad w_{\beta j} - w_{\beta i} \ge \frac{1}{2}\sum_{\alpha=1}^{3} \left(L_{\alpha i}\delta_{\alpha\beta i} + L_{\alpha j}\delta_{\alpha\beta j}\right) - \left(1 - \sigma^{-}_{\beta ij}\right) D_\beta \tag{4-2}$$

$$\forall i, \forall j, i < j \quad \sum_{\beta=1}^{3} \left(\sigma^{+}_{\beta ij} + \sigma^{-}_{\beta ij}\right) \ge \chi_i + \chi_j - 1 \tag{5}$$

where $\sigma^{+}_{\beta ij}, \sigma^{-}_{\beta ij} \in \{0, 1\}$.

The condition $\sigma^{+}_{\beta ij} = 1$ implies that the side's projections of i and j do not overlap along the w_β axis and j precedes i, while $\sigma^{+}_{\beta ij}$ =0 makes the corresponding *non-intersection* constraint redundant. Analogous considerations hold for $\sigma^{-}_{\beta ij}$. Equation (5) guarantees that, if both i and j are picked, at least one of the *non-intersection* (*big-M*) constraints (4) holds. (It can be easily proved that the terms D_β are the minimum *big-Ms* for constraints 4-1 and 4-2 allowing, $\forall i, j, i < j$, all *feasible configurations* of *i,j* in D).

The *objective function* has the following expression:

$$\max \sum_{i=1}^{n} K_i \chi_i \tag{6}$$

where K_i is the volume or mass of item i.

The model adopted by Pisinger and Sigurd (2005, Sect. 2) is based on the formulation given by Chen et al. (1995) and Onodera et al. (1991). The model proposed by Chen is more general, considering the 3-dimensional case, with orthogonal rotations. When just a container is considered, the model of Chen (excluding conditions 8 and 9) reduces to that of the *Single bin 3D-Packing* (*Basic Problem*). Conditions (4-1), (4-2) and (5) of the *Basic Model* described here (7 conditions for each pair of items) correspond to conditions (1)–(7) of Chen's model. In the model proposed by Chen the bottom left vertex is considered as origin of the local reference frame of each item. In the *Basic Model* formulated here, it is centered in its geometrical center (assumed coincident with its center of mass). This gives a simple form to the terms $\sum_{i=1}^{n} M_i w_{\beta i}$ introduced for the balancing conditions (see Fasano 2004 and conditions 7–8, Sect. 2.2, of the present article). By conditions (1)–(3) the contribution of the (*static moments*) $M_i w_{\beta i}$ are null, when $\chi_i = 0$. (If $\chi_i = 0$ would not imply $w_{\beta i} = 0$, the *static moments* should be formulated indirectly by means of auxiliary constraints, e.g., by introducing further *big-M* constraints, since the terms $\chi_i M_i w_{\beta i}$ would not be linear anymore).

A *Basic Model* instance relative to *n* items contains: O(3n(n-1)), (binary) variables of type σ, O(9n) of type δ, $O(n)$ of type χ, O(6n) orthogonality constraints, O(3n) domain constraints, O(3n(n-1)+n(n-1)/2) non-intersection constraints (of which O(3n(n-1)) big-M constraints).

Remark 1 Items with pre-fixed orientation/position can be treated very easily, simply by fixing their orientation (δ *variables)* and/or their center coordinates. The variables of type δ correspond to the item directions cosines for orthogonal rotations ($\pm\pi/2$).

Remark 2 In the *Basic Model*, it is assumed that the item center of mass and its geometrical center are coincident. This assumption simplifies significantly the formulation of the model and it is usually acceptable in practice. In this case there are six possible rotations. However, if the item is unsymmetrical, 24 rotations have to be considered. This is the case of *tetris*-like items (see Fasano 2004 and Sect. 2.2). An item consisting of a single non-homogeneous parallelepiped, can be simply considered as composed by two elements: one parallelepiped with zero mass, geometrically identical to the item itself, and a point mass, with the item mass and its position coincident with the item center of mass. The composed item can then be treated as a (degenerate) *tetris*-like item, whose components are the parallelepiped with zero mass and the point mass. The formulation used for the rotation of the *tetris*-like items (see Fasano 2004 and Sect. 2.2) could also be adopted for single homogeneous parallelepipeds, limiting the rotations to 6.

2.2 Extensions

When dealing with the classical *Three-dimensional (Single) Bin Packing* Problem, the *modelling* (*non-algorithmic*) approach proposed in Sect. 2.1 is generally less efficient than other methods reported in the literature (Martello et al. 2000; Pisinger 1998). The MIP approach seems however quite suitable to tackle a wide class of non-standard packing issues with additional constraints, arising frequently in practice. Some possible extensions, including *balancing* conditions, *tetris*-like items and non-standard domains are shortly overviewed in this section, referring to Fasano (2003, 2004) for a deeper analysis.

Static balancing conditions consist of imposing the overall center of mass must stay within a given convex domain (a *polytope* or approximated by a *polytope*) of vertices
$\boldsymbol{V_1}(V_{11}, V_{21}, V_{31}), \ldots, \boldsymbol{V_r}(V_{1r}, V_{2r}, V_{3r})$. In Fasano (2004) it has been shown that the *static balancing* can be easily expressed by the following linear conditions:

$$\sum_{\gamma=1}^{r} \psi_{\gamma}* = m \tag{7}$$

$$\forall \beta \quad \sum_{i=1}^{n} M_i w_{\beta i} = \sum_{\gamma=1}^{r} V_{\beta\gamma} \psi_{\gamma}* \tag{8}$$

where $m = \sum_{i=1}^{n} M_i \chi_i, \forall \gamma, \psi_{\gamma}* \geq 0$.

(It is supposed $m > 0$. Notice that by conditions 3, for each non-picked item i, all $w_{\beta i}$ variables are null).

The above balancing conditions are simplified when C is rectangular. In this case they have the simple form: $\forall \beta\ C_{L\beta}\ m \leq \sum_{i=1}^{n} M_i w_{\beta i} \leq C_{U\beta} m$, where, $\forall \beta$, [$C_{L\beta}$, $C_{U\beta}$] are the admitted intervals for the *overall* center of mass (relative to all picked items).

***Tetris*-like items**, consisting of clusters of mutually orthogonal parallelepipeds (see Fig. 2), can be considered. The following equations are posed, for each (*tetris*-like) item i and each component h of it:

$$\forall i \quad \sum_{\omega} \vartheta_{\omega i} = \chi_i \tag{9}$$

$$\forall \beta, \forall\ h \in C_i, \forall i\ w_{\beta h i} = o'_{\beta i} + \sum_{\omega} W'_{\omega \beta h i} \vartheta_{\omega i} \tag{10}$$

The binary variable $\theta_{\omega i}$ is introduced with the following meaning: $\theta_{\omega i} = 1$ if item i (is picked and it) has the orientation ω, $\theta_{\omega i} = 0$ otherwise (in the 3D-case there are 24 possible orientations ω).

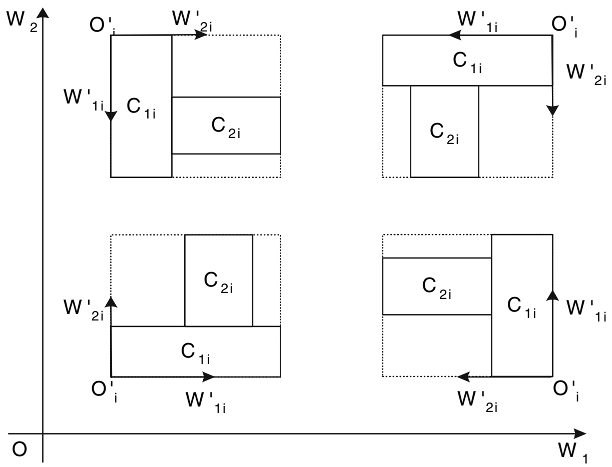


Fig. 2 *Tetris*-like item (two-dimensional representation)

Equation (9) generalizes (1) and (2). In (10) C_i indicates the set of components associated to each item i, $W'_{\omega\beta hi}$ is the distance between the (center) coordinates of component h and the origin of the *local* reference frame, along the w_β axis of the *overall* reference frame, corresponding to the orientation ω . The terms $w_{\beta hi}$ and $o'_{\beta hi}$ are the coordinates of the h component and of the local reference frame .

Non-intersection constraints (4) can be generalized as follows, for each component *h, l* of (*tetris*-like) items i and j:

$$\forall\beta, \forall h \in C_i, \forall l \in C_j, \forall i, \forall j, i < j$$
$$w_{\beta hi} - w_{\beta lj} \geq \frac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i} + L_{\omega\beta lj}\vartheta_{\omega j}\right) - \left(1 - \sigma^{+}_{\beta hlij}\right)D_\beta \quad \text{(11-1)}$$
$$\forall\beta, \forall h \in C_i, \forall l \in C_j, \forall i, \forall j, i < j$$
$$w_{\beta lj} - w_{\beta hi} \geq \frac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i} + L_{\omega\beta lj}\vartheta_{\omega j}\right) - \left(1 - \sigma^{-}_{\beta hlij}\right)D_\beta \quad \text{(11-2)}$$

where $L_{\omega\beta hi}, L_{\omega\beta lj}$ are the sides of the h, l components, parallel to the w_β axis, corresponding to the orientation ω of items i and j, respectively.

Domain-conditions (3) (properly modified by replacing the terms $\delta_{\alpha\beta i}L_{\alpha i}$ with $L_{\omega\beta hi}\theta_{\omega i}$) can be applied to each single component of each item and the generalization of condition (5) is straightforward.

Non-standard domains can be considered as further extension of the *Basic Model*. **Convex** (non-rectangular) **domains** (polytopes) can be treated by imposing each vertex p, of each parallelepiped i, must stay within the given convex domain D^*. Denoting by $\boldsymbol{V_1}(V_{11}, V_{21}, V_{31}), \ldots, \boldsymbol{V_u}(V_{1u}, V_{2u}, V_{3u})$ the u vertices of D^*, (*convexity*) conditions of the following form hold for each vertex p of each item i:

$$\forall\beta, \forall i, \forall p \quad w_{\beta i} \pm \frac{1}{2}\sum_{\alpha=1}^{3} L_{\alpha i}\delta_{\alpha\beta i} = \sum_{\gamma=1}^{u} V_{\beta\gamma}\psi_{\gamma ip} \quad (12)$$

$$\forall i, \forall p \quad \sum_{\gamma=1}^{u}\psi_{\gamma ip} = \chi_i \quad (13)$$

where $\forall\gamma, \forall i, \forall p, \psi_{\gamma ip} \geq 0$.

The *Domain* constraints (3) are then substituted by (12) and (13). Rectangular (or *tetris*-like) ***holes*** within the domain can be immediately tackled by fixing (position and orientation) of items with zero mass (i.e., forbidden regions, see Fig. 3). Separation planes (parallel to a base of the domain and with their position variable within a given range, see Fig. 3) can simply be considered as zero-mass parallelepipeds with a zero dimension. (An alternative formulation is reported in Fasano 2003).

In the following, the *Basic Problem* or any extension of it, susceptible to a MIP formulation, are simply denoted as *3D-packing Problem* and analogously any (MIP) extension of the *Basic Model* is denoted as *3D-packing Model*.

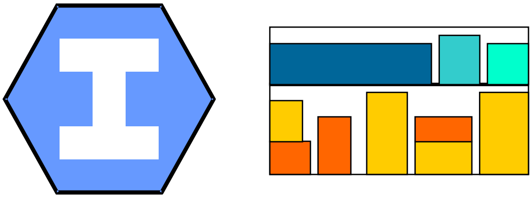


Fig. 3 Non-standard domains (two-dimensional representation)

2.3 Intrinsic difficulties

The presence, in the *3D-packing Model*, of a very large number of *big-M* constraints, essentially related to the *non-intersection* conditions, represents a major intrinsic difficulty (an instance with 100 items involves about 3×10^4 *non-intersection linearized logical* constraints). Furthermore, when dealing with real world instances, the model contains a huge number of implicit conditions (Fasano 2004) that make the task of finding mutually compatible σ values extremely arduous. For instance, $\forall i, j$ the following implicit conditions hold:

if $L_{1i} + L_{1j} \geq D_\beta$ *, then items i and j cannot be aligned along the* w_β *axis;*

if $L_{1i} + L_{2j} \geq D_\beta$ *, then items i and j cannot be aligned along the* w_β *axis, with* L_{2j} *parallel to the* w_β *axis.*

Less trivial implicit *alignment/orientation* implications could involve larger subsets of items such as triplets, four-tuples and so on, giving rise to implicit (necessary) conditions on the σ (and δ) variables. The following *transitivity* implicit constraints hold moreover for each triplet of items *i,j,k* (Fasano 2004; Fekete and Schepers 2004, 2006; Pisinger and Sigurd 2005):

if i precedes j and j precedes k along the w_β *axis, then i precedes k along the same axis* . (Notice that *transitivity* constraints hold independently from the domain and items dimensions). Non-zero σ values compatible with the *transitivity* constraints reported above are called *transitivity*-compatible.

As previously pointed out (Fasano 2004), a number of *auxiliary* constraints on the σ (and δ) variables could be profitably introduced in the *3D-packing Model*, on the basis of the above implicit implications (and others reported in Fasano 2004). It is well known, in fact, that a MIP model is frequently made easier to solve by introducing *valid inequalities* to tighten the *LP-relaxation*. In particular, the following *transitivity auxiliary* constraints on the σ variables can be introduced:

$$\forall \beta, \forall i, \forall j, \forall k, i < j < k \quad \sigma^-_{\beta ik} \geq \sigma^-_{\beta ij} + \sigma^-_{\beta jk} - 1 \tag{14}$$

Analogous *auxiliary* constraints are generated by all possible permutations of *(i,j,k)*. The introduction of *auxiliary* constraints could however increase dramatically the scale of the resulting MIP model (e.g., in instances of 100 items the *transitivity auxiliary* constraints are about 3×10^6) and a *Branch and Cut* approach could be advantageously adopted (Padberg 1999).

An alternative philosophy is proposed in this paper (Sect. 3) to efficiently look into satisfactory (sub-optimal) solutions of the *3D-packing Model*. It consists essentially in a heuristic procedure based on the (recursive) generation of sets of *transitivity-compatible* σ variables.

3 MIP-based heuristic

3.1 Basic concepts

The proposed heuristic is based on the concepts of *relative positions* and *abstract configuration*, defined below.

Definition 1 (Relative position constraints)
Constraints of the form:

$$w_{\beta i} - w_{\beta j} \geq \frac{1}{2}\sum_{\alpha=1}^{3}\left(\delta_{\alpha\beta i} L_{\alpha i} + \delta_{\alpha\beta j} L_{\alpha j}\right) \tag{15-1}$$

$$w_{\beta j} - w_{\beta i} \geq \frac{1}{2}\sum_{\alpha=1}^{3}\left(\delta_{\alpha\beta i} L_{\alpha i} + \delta_{\alpha\beta j} L_{\alpha j}\right) \tag{15-2}$$

$i, j \in \{1, \ldots, n\}, i < j, \beta \in \{1, 2, 3\}$ are called relative position constraints with respect to the w_β axis (the constraints 15-1 and 15-2 are mutually exclusive).

Definition 2 (Abstract configuration)
Given n items, a set of n(n-1)/2 relative position constraints, one and only one, $\forall i, j, i < j$, is an abstract configuration if the associated feasibility region is not empty.

Each *abstract configuration* corresponds to a sub-set of *non-intersection* constraints (4).The whole heuristic procedure is based on the generation of *abstract configurations*. It is immediate to see that an *abstract configuration* corresponds to a set of *transitivity*-compatible σ variables (and vice-versa).

In an unbounded domain, any *abstract configuration* would determine a feasible solution (with no intersection) for all n items and, as illustrated by Fig. 4 (2D-representation), given any *abstract configuration*, items can be translated or rotated maintaining their *relative position*.

A grid of the type depicted in Fig. 5 (2D example with three items; $L_1^* = \min_i\{L_{1i}\}$, $N_x = \text{int}\{D_x/L_1^*\}, N_y = \text{int}\{D_y/L_1^*\}$) could in principle be adopted to generate all *abstract configurations* (simply by neglecting the actual dimensions of items and associating each item to a grid node, in all possible ways).

The proposed procedure aims at generating a sequence of *good abstract configurations* and at solving, step by step, a reduced *3D-packing Model* obtained by eliminating all the redundant *non-intersection* constraints (i.e., *non-intersection* constraints not corresponding to the current *abstract configuration*). At each step, items are rejected, if necessary, to make the current *abstract configuration* compatible with the given domain D.

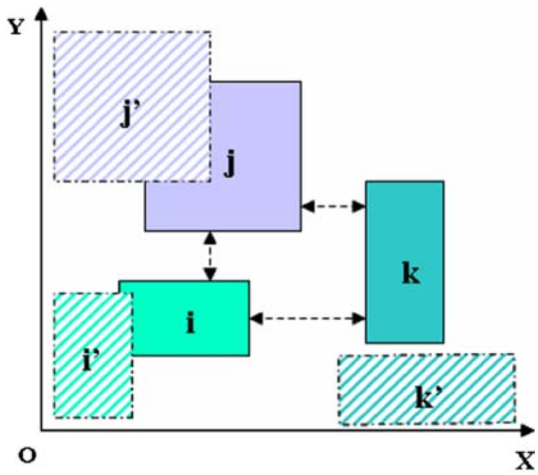


Fig. 4 Example of translations/rotations compatible with a given *abstract configuration*

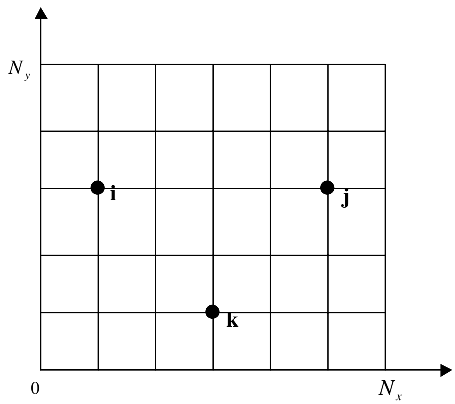


Fig. 5 Example of *abstract configuration* generation grid

3.2 Procedure logic

This section describes the heuristic procedure overall logic. Its concept is based on the *abstract configurations* generation that acts on the *non-intersection* conditions, independently from any possible non-standard conditions. It is thus applicable to any non-standard problem, provided that it is susceptible to a MIP formulation.

A toy example is introduced here to better illustrate the heuristic overall logic. Even if the procedure logic is not affected by the presence of non-standard conditions, the toy example refers to a (realistic) non-standard packing problem, to better fit into the overall context of this work.

The considered domain is a box (parallelepiped) containing two internal structural supports (parallelepipeds), with fixed positions, acting as forbidden regions (see Fig. 6). A separation plane, parallel to the basis of the domain is present. Its position (with respect to the basis) is not fixed and varies within a given range. The overall center of mass must stay in a given squared domain, centered with respect to the container.

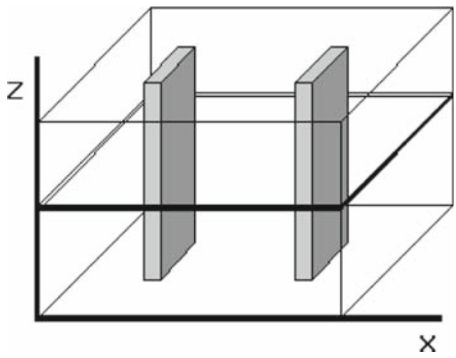


Fig. 6 Toy example domain

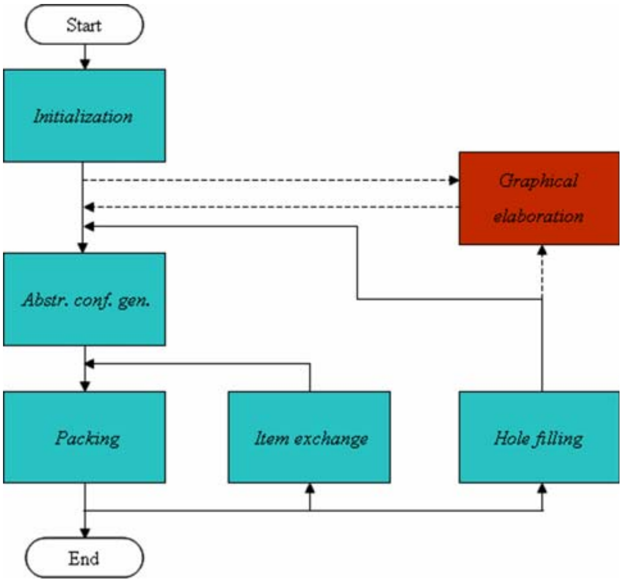


Fig. 7 Heuristic high level logic

The proposed heuristic is based on the following modules: *Initialisation, Abstract Configuration Generation, Packing, Hole-filling, Item-exchange*. Figure 7 illustrates the heuristic high level logic (and a possible interaction with a graphical system). The procedure modules are illustrated in the following.

Initialization The goal of this module is to obtain a *good* approximate initial solution that takes into account both the basic and non-standard conditions (e.g., *static balancing*, separation planes, prefixed orientation/position of some items) of the *3D-Packing Problem* to solve. An *LP-relaxation* of the σ variables is performed dropping their integrality conditions and an ad hoc objective function (Fasano 1999) aimed at *minimising* the intersection between items is adopted (together with a reformulation of constraints (4), see Fasano 1999). In this phase intersections between items are

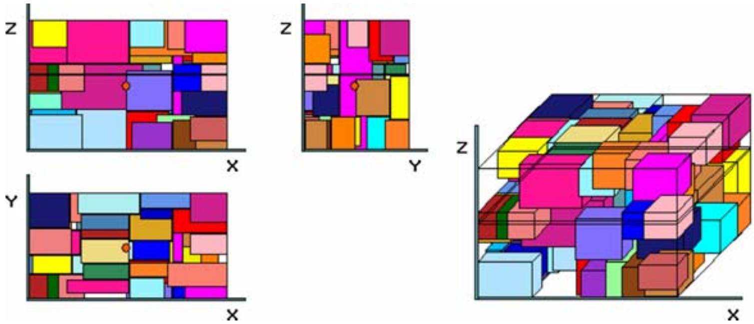


Fig. 8 Toy example initial solution

admitted, while all the other basic and additional constraints must be respected. All the given items can be considered or just a subset of them and all corresponding χ variables are set to one. If the δ variables are not fixed the resulting problem is still a MIP one and the solution process is stopped as soon as a first integer solution is found.The items orientation could however also be fixed, reducing the whole model to an LP one. This could be done, for instance, imposing that the smallest side of each item is parallel to the smallest side of the domain and that the biggest side of each item is parallel to the biggest side of the domain. The items orientation could be subsequently changed by the *Packing* module, for which the δ variables are free. (A number of auxiliary constraints of the kind reported in Sect. 2.3 can also be profitably introduced to tighten the *LP-relaxation* of the non-intersection constraints).

The (approximate) solution so obtained is given as input to the *Abstract Configuration Generation* module. Figure 8 represents an initial solution for the toy example (with 75 items). The point marker indicates the *overall* center of mass actual position. In the toy example initial solution, a number of items are overlapping or crossing the separation plane (as admitted in this phase).

Remark 3 The introduction of an ad hoc objective function has been described at a quite detailed level in Fasano (1999) and the reader is referred there for a deeper analysis. The *non-intersection* constraints (4) of the *Basic Model* are reformulated in an *LP-relaxed* form. The ad hoc objective function minimizes the overall overlapping of the items projections and thus, strictly speaking, it is a surrogate of the objective function that would *minimize* the *overall overlapping* volume.

Abstract configuration generation This module aims at generating an *abstract configuration*, starting from an approximate solution (obtained by the *Initialisation/Hole-filling* modules or even by a graphical elaboration). A *relative position* constraint is derived for each pair of items. For non-intersecting items, the satisfied *relative position* constraints are considered. When more than one *relative position* constraint is satisfied for the same pair of items, the one corresponding to the maximum relative distance between the coordinates of items is selected. For intersecting items, the *relative position* corresponding to the maximum relative distance between the coordinates of items is chosen. The generated *abstract configuration* is given as input to the *Packing* module.

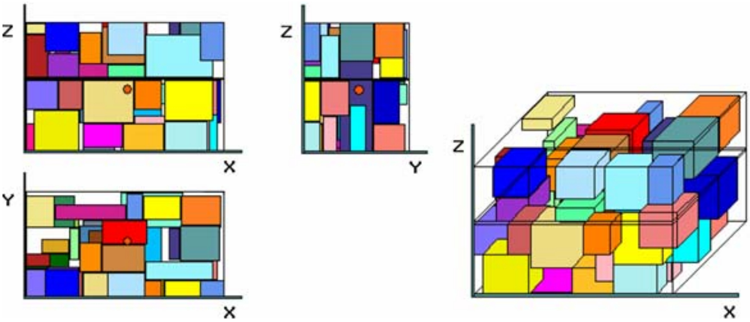


Fig. 9 Toy example *Packing* module solution

Packing The goal of the *Packing* module is to look for a solution to the (original) *3D-packing Problem*, relative to a given *abstract configuration*. The *non-intersection* constraints (4) corresponding to the (current) *abstract configuration* are maintained (a single one $\forall i, j, i < j$), while all the remaining *non-intersection* constraints (4) are eliminated. In (5) each σ variable corresponding to the current *abstract configuration* is maintained, while the other eliminated (the integrality condition on the single σ variable in 5 can be dropped). The resulting *3D-packing Model* is then solved adopting a Branch and Bound approach. The binary variables χ, σ and δ are processed sequentially by groups of items, ordered by volume (or mass), following a *depth first* strategy (during the search subsets of binary variables can be temporarily fixed). A lower *cut-off* can be set on the basis of the best-so-far solution and (part of the) items previously picked can be imposed, following a *greedy* approach. If a satisfactory solution is found, it is taken as final solution and the whole process ends. Otherwise, the best-so-far solution is stored and the process continues by activating the *Hole-filling* or *Item-exchange* modules (or even by interacting graphically). A stopping rule (e.g., on the maximum number of iterations) can be introduced. Figure 9 shows the *Packing* module solution obtained from the toy example initial solution (a number of empty spaces are visible).

Remark 4 The *Packing* moldule utilizes a reduced *3D-packing Model* that is still a general MIP model (i.e., *NP hard*). One and only one *non-intersection* constraint (4) is taken into account $\forall i, j, i < j$, with its relative σ variable. It is thus quite evident that the introduction of an *abstract configuration* determines a dramatic reduction of the original *3D-packing Model*.

Hole-filling This module is aimed at performing a *non-blind* local search by perturbing the *Packing* module (current) solution. Empty spaces are exploited, whenever possible, to obtain an improved approximate solution (better in terms of volume or mass loaded, but with possible intersections) and a hopefully improved subsequent *abstract configuration*. The *Packing* module (current) solution is *immersed* into a *grid* domain (see Fig. 10) and a number of non-picked items are pre-selected as candidates to cover non-covered nodes of the *grid*.

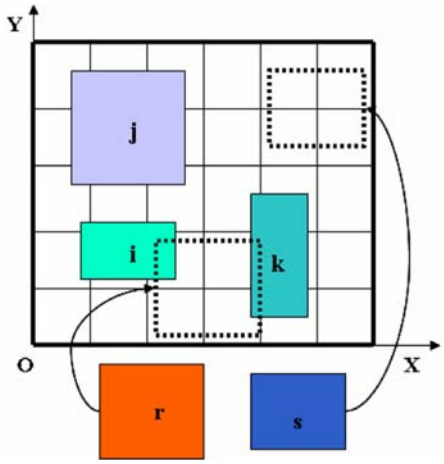


Fig. 10 *Hole-filling* concept (two-dimensional representation)

An ad hoc *allocation* MIP model is utilized to maximize the loaded volume (or mass), adding (if possible) new items (from the pre-selected candidates). The binary variable $\xi_{s\nu}$ is associated to each pre-selected item s and to each non-covered node ν, with $\xi_{s\nu} = 1$ if item s is allocated to node ν (with its center coincident with node ν) and $\xi_{s\nu} = 0$ otherwise. (For each item s only nodes allowing the item can be contained, with a proper orientation, within the domain are considered). The (allocation) equation below is introduced:

$$\forall \beta, \forall s \quad w_{\beta s} = \sum_{\nu} U_{\beta\nu}\xi_{s\nu} \tag{16}$$

where $U_{\beta\nu}$ are the coordinates of the (available) nodes. It assignes the location of item s on the grid (when picked). The following equation guarantees that at most one item s is allocated to the same node:

$$\forall \nu \quad \sum_{s} \xi_{s\nu} \le 1 \tag{17}$$

Conditions (1)–(3) are posed for all item s. All items already picked (corresponding to the *Packing* module current solution) are fixed. Since in this phase intersections are admitted, conditions (4) are not included in the model. All additional constraints (of the original *3D-packing Model*) are, on the contrary, contemplated in the model. This is the case, for instance, of the *balancing* constraints. The actual positioning of the new items (when admissible) is carried out, together with the previously picked items, in a subsequent phase, by the *Packing* module that considers all the constraints, including the *non-intersection* ones. The approximate solution obtained by the *Hole-filling* module is given as input to the *Abstract configuration generation* module.

Remark 5 Prior to activating the *Packing* module (once the new items have been chosen), a post-optimization could be performed, to *minimize* the items *overall overlapping* (in the sense clarified in *Remark 3*) by reallocating them on the grid, with an approach similar to that adopted by the *Initialization* module. (The items deriving from the *Packing* module solution can be translated and/or rotated on the basis of their *abstract configuration).*

Item-exchange This module is aimed at performing a *non-blind* local search by perturbing the *abstract configuration* relative to the current *Packing* solution (to tentatively give rise to an improved *abstract configuration*). Items are exchanged (in the current *abstract configuration*) to increase the loaded volume (or mass). Picked items are exchanged with *bigger* non-picked items (or with items with bigger mass, if the loaded mass is maximised). Non-picked items can also be exchanged with *smaller* non-picked items (or items with smaller mass, if the loaded mass is maximised). Exchanges likely to be advantageous in terms of loaded volume (or mass) are performed, without taking into account (explicitly) the constraints of the *3D-packing Model* to solve that are contemplated in a subsequent phase, by the *Packing* module.

Depending on the adopted strategy, this module, by exchanging a (limited) number of items, accomplishes either a *weak* or a *strong* perturbation of the current *abstract configuration*. When a *weak* perturbation strategy is adopted, the exchanged items are not too different (in terms of mass and volume) from each other and, on the contrary, quite different, when a *strong* perturbation strategy is chosen. When a *weak* perturbation is performed, however, the new *abstract configuration* remains in great measure *close* to the original one. (The corresponding solution, being a *weak* perturbation of the previous one, which is feasible, is quite likely to be a good approximation of an improved feasible solution. The *3D-packing Model* constraints are so indirectly considered, through the *neighbourhood* with the initial solution or the *Packing* module one). The (approximate) solution obtained by the *Item-exchange* module is given as input to the *Packing* module.

Remark 6 If the exchange of the picked item s with the non-picked item s' is not feasible, the *Packing* module drops item s'. It is however possible to avoid this. Instead of generating a new *abstract configuration* by forcing the exchange $s - s'$ it could be sufficient to duplicate for the items s' all the relative positions corresponding to s and pose, in the *Packing* module the condition:

$\chi_s + \chi_{s'} = 1$ (updating subsequently the *abstract configuration* on the basis of the obtained *Packing* solution).

The *Hole-filling* and *Item-exchange* modules can be activated in various sequences, following different *item-selection* (/optimization) strategies. The whole procedure can be restarted (recursively) from the *Initialization* module, forcing the *abstract configuration*, relative to subsets of items already picked. The *Initialization* module itself can be used recursively in conjunction with the *Abstract Configuration Generation* module. The proposed heuristic tackles the *3D-packing Problem* recursively, reducing dramatically, at each step, the difficulties related, to the *non-intersection* (*big-M*) constraints (4) of the (original) *3D-packing Model* to solve. By the *Hole-filling* and *Item-exchange* modules this procedure performs a *non-blind* local search. The actions activated by these modules are indeed oriented to increase the loaded volume (or mass), taking into account, approximately (or indirectly), the given packing constraints, except for the *non-intersection* ones that are explicitly treated by the *Packing* module.

3.3 Experimental analysis

The packing problems considered in this work, just because non-standard, are quite difficult to classify, as well as it is quite difficult to perform statistics on them. The efficiency of the approach proposed depends on a variety of factors, since the difficulties of the problems to solve (in addition to the number of items involved) are indeed strongly dependent on the items and domain characteristics, as well as on overall constraints such as the *balancing* ones. The separation planes, as it is self-evident, reduce significantly the volume exploitation, especially when the items are quite different from each other, in terms of volumes, dimensions, and ratios between their dimensions. The presence of *tetris*-like items also makes the problem more difficult to solve. A rough evaluation of the impact due to the presence of *tetris*-like items can be obtained considering the total number of single parallelepipeds and *tetris*-like components. The situation seems much simpler, from the experimental analysis point of view, when the problem concerns the packing of parallelepipeds into a parallelepiped, in the presence of the sole (*static*) *balancing*. The difficulties related to the center of mass domain tightness are however not independent from the distribution of the item typologies (in terms of mass, volume, dimensions). Roughly speaking, it could be said that some percentage of admissible off-centering (with respect to the container dimensions) can decrease by 15–20% the exploited volume and increase by up 25–30% the computational effort. These estimates are however very imprecise and indicate just a general trend.

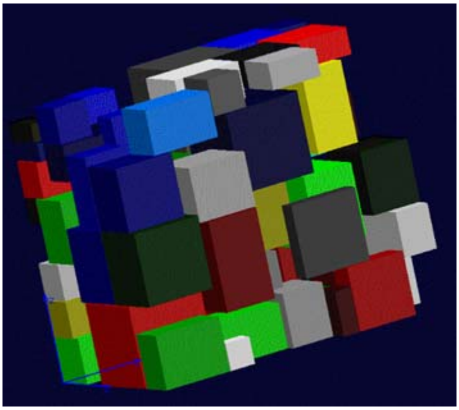
An experimental analysis has been performed considering more than 100 case studies involving up to 100 parallelepipeds, with a parallelepiped as domain and subject to the (sole) *static balancing* conditions. The IBM OSL V3 (2001) has been used as LP/MIP solver, with Windows XP Professional as operating system. The following hardware has been adopted: Pentium 4, 2.8 GHz CPU, 512 Mb RAM, 74 Gb HD. A survey is reported in Table 1.

A case study, concerning the packing of parallelepipeds into a parallelepiped with (*static*) *balancing* constraints (center of mass rectangular domain) is illustrated in Fig. 11. (The instance involves 100 items, the *static balancing* admissible error is ± 1 unit with respect to each axis and the domain dimensions are 21,35,30 units, respectively). The performed optimisation strategy is given by the sequence: (1) *Initialisation*, (2) *Packing*, (3) *Initialisation*, (4) *Packing*, (5) *Hole filling*, (6) *Packing*, (7) *Item exchange*, (8) *Packing*. The process has been stopped by the user after $\sim$ 300

Table 1 Experimental analysis

Module	Involved items	CPU time estimates (s)
Initialization	75–100	45–90 (recursive mode)
Abstract configuration generation	75–100	< 5
Packing	75–100	30–60
Hole-filling	10–15	< 15
Item exchange	10–15	< 5

Fig. 11 Case study balanced solution



CPU seconds and the obtained packing (∼80 picked items; ∼70% exploited volume) is depicted in the figure.

Remark 7 The proposed approach has been successfully utilized (by the CAST system, see Sect. 1) in the real world frame of the of the *cargo accommodation* analysis for the generation of the ATV Jules Verne cargo layout (January 2005, July 2005, February 2006 and July 2006), to pack small items into bags (about 200). In most cases, in fact, the presence of separation planes has to be considered, together with the fixed position or orientation of specific items, as well as tight *static balancing* conditions. (The introduction of *tetris*-like items is at present under implementation for the accommodation of large items on the rack fronts). CAST is expected to support the whole analytical cargo accommodation for all future ATV missions, starting from 2007.

4 Conclusive remarks

This paper originates from a research activity performed by Alcatel Alenia Space Italia, in support to the *cargo accommodation* of space vehicles and modules. In a previous work, it has been shown that mixed integer programming is quite suitable to formulate a wide class of *non-standard* three-dimensional packing problems, including *tetris*-like items and non-standard domains, with additional conditions, such as the *static balancing*. The paper overviews the MIP approach focusing on the *Three-Dimensional Single Bin Packing* problem (denoted here as *Basic Problem*). A MIP-based heuristic is introduced to tackle efficiently the *3D-packing Problem*, consisting of the *Basic Problem* with any possible extension, susceptible to a MIP formulation (*non-standard* items/domains and additional conditions).

The heuristic consists of a recursive procedure based on a *non-blind* local search philosophy. The concept of *abstract configuration*, concerning the *relative positions* between items, is introduced: the *relative positions* of items, determined by any *abstract configuration*, give rise to a feasible solution in an unbounded domain. The heuristic generates a sequence of *good abstract configurations*. At each step, on the basis of the current *abstract configuration*, all redundant *non-intersection* (*big-M*) constraints (together with the corresponding binary variables) are eliminated. The resulting MIP model is drammatically simplified, being the *non-intersection big-M* a major intrinsic difficulty of the *3D-packing* problem.

The paper describes the heuristic modules and the procedure overall logic.

The modules can be activated in different sequences, following different optimization strategies. Items can be added step by step (allowing a *greedy* approach) and the process can start or restart from any approximate solution. This makes the heuristic quite suitable to be used in combination with a graphical system, enabling also the user to interact with the whole optimization process at any time by suggesting *partial* solutions.

An experimental analysis has been performed for a number of case studies.

Further applications of the proposed approach could be considered in several fields, not limited to space engineering only. A dedicated future activity could focus on the comparative experimental analysis of different optimization strategies, as well as on the selection and tuning of the most promising ones. Applications to different Operations Research areas (e.g., scheduling/re-scheduling problems) could also be considered.

References

Addis B, Locatelli M, Schoen F (2006) Efficiently packing unequal disks in a circle: a computational approach which exploits the continuous and combinatorial structure of the problem. Optimization Online (http://www.optimization-online.org) 1343

Blazewicz J, Hawryluk P, Walkowiak R (1993) Using a tabu search approach for solving the two-dimensional irregular cutting problem. AOR 41:313–327

Castillo I, Kampas FJ, Pintér JD (2006) Solving circle packing by global optimization: numerical results and industrial application. (Submitted)

Coffman E, Garey JM, Johnson D (1997) Approximation algorithms for bin packing: a survey. PWS Publishing Company, Boston

Chen CS, Lee SM, Shen QS (1995) An analytical model for the container loading problem. Eur J Oper Res 80:68–76

Daughtrey RS et al (1991) A simulated annealing approach to 3D packing with multiple constraints, Cosmic Program MFS28700, Boeing Huntsville AI Center Huntsville (Alabama)

Dyckhoff H, Scheithauer G, Terno J (1997) Cutting and packing. In: Dell'Amico M et al. (eds) Annotated bibliographies in combinatorial optimization, Wiley, Chichester

Egeblad J, Nielsen BK, Odgaard A (2006) Fast neighborhood search for two- and three-dimensional nesting problems. Euro J Oper Res (in press)

Fasano G (1989) Satellite optimal layout IBM Europe Institute 1989, application of mathematical and optimization techniques, Garmisch PK (Ge)

Fasano G (1999) Cargo analytical integration in space engineering: a three-dimensional packing model. In: Ciriani T, Gliozzi S, Johnson EL (eds) Operations research in industry, Macmillan, pp 232–246

Fasano G (2003) MIP Models for solving three-dimensional packing problems arising in space engineering. In: Ciriani T, Fasano G, Gliozzi S, Tadei R (eds), Operations research in space and air. Kluwer, Boston, pp 43–56

Fasano G (2004) A MIP approach for some practical packing problems: balancing constraints and tetris-like items. 4OR 2:161–174

Fasano G et al (2003) A Cargo accommodation problem for a space vehicle: the CAST project In: Ciriani T, Fasano G, Gliozzi S, Tadei R (eds) Operations research in space and air, Kluwer, Dordrecht, pp 13–26

Fekete S, Schepers J (2004) A combinatorial characterization of higher-dimensional orthogonal packing. Math Oper Res 29:353–368

Fekete S, Schepers J (2006) An exact algorithm for higher-dimensional orthogonal packing. Oper Res (in press)

Fischetti M, Luzzi I (2003) Exact and heuristic MIP models for nesting problems. EURO XIX Conference, Istanbul, Turkey

Gones AM, Oliveira JF (2002) A 2-exchange heuristics for nesting problems. Eur J Oper Res 141:359–570

Martello S, Pisinger D, Vigo D (2000) The three-dimensional bin packing problem. Oper Res 48:256–267

Martello S, Pisinger D, Vigo D, Edgar den Boef Korst J (2006) Algorithms for general and robot-packable variants of the three- dimensional bin packing problem. ACM Trans Math Software (in press)

Mathur K (1998) An integer programming-based heuristic for the balanced loading problem. Oper Res Lett 22:19–25

Oliveira JF, Ferreira JS (1993) Algorithms for nesting problems. In: René Vidal VV (ed) Applied simulated annealing. Lecture Notes in Economics and Mathematical Systems, Springer, Heidelberg, pp 255–273

Onodera H, Taniguchi Y, Tmaru K (1991) Branch-and-bound placement for building block layout. 28th ACM/IEEE design automation conference, pp 433–439

Padberg M (1999) Packing small boxes into a big box. New York University, Office of Naval Research, N00014-327

Pintér JD, Kampas FJ (2004) Generalized circle packings: model formulations and numerical results. In: Proceedings of the 6th international mathematica symposium, Banff, AB, Canada

Pintér JD, Kampas FJ (2005) Nonlinear optimization in mathematica with mathoptimizer professional. Math Educ Res 10(2)

Pisinger D (1998) A tree search heuristic for the container loading problem. Ric Oper 28:31–48

Pisinger D, Sigurd MM (2005) The two-dimensional bin packing problem with variable bin sizes and costs. Disctrete Optim 2:154–167

Pisinger D, Sigurd MM (2006) Using decomposition techniques and constraints programming for solving the two-dimensional bin packing problem. INFORMS J Comput (in press)

Stoyan Y et al (2004) Φ-functions for complex 2D-objects, 4OR 2:69–84

Tadei R et al (2003) A heuristic procedure for the space cargo rack configuration problem. In: Ciriani T, Fasano G, Gliozzi S, Tadei R (eds) Operations research in space and air, Kluwer, Dordrecht, pp 27–42

Takadama K et al (2004) Capabilities of a multiagent-based cargo layout system for H-II transfer vehicle, 16th IFAC symposium on automatic control in aerospace