J Glob Optim
DOI 10.1007/s10898-012-9865-8

A global optimization point of view to handle non-standard object packing problems

Giorgio Fasano

Received: 8 September 2010 / Accepted: 27 January 2012

Abstract This work originates from research carried out in support to the cargo accommodation of space vehicles/modules. The goal of this activity is to maximize the loaded cargo, taking into account the given accommodation requirements. Items can often be modelled as parallelepipeds, but it is even more frequent that real-world issues make this approximation no longer acceptable. These aspects and the presence of additional overall conditions, such as balancing, give rise to very challenging non-standard packing problems, not only in the frame of space engineering, but also in different application areas. This article considers first the orthogonal packing of *tetris*-like items, within a convex domain and subsequently the packing of polygons with (continuous) rotations in a convex domain. The proposed approach is based on mixed integer linear/non-linear programming (MIP, MINLP), from a global optimization point of view. The *tetris*-like formulation is exploited to provide the MINLP solution process with an approximated starting solution.

Keywords Tetris-like item packing · Non-rectangular domains · Polygon packing · MIP/MINLP

1 Introduction

This work originates from research carried out in a space engineering frame, in support to the cargo accommodation of space vehicles and modules. The goal of this activity is to maximize the loaded cargo (in terms of volume or mass), in compliance with the given accommodation rules and requirements. Very complex geometrical aspects have to be taken into account together, in addition to balancing conditions, deriving from very tight attitude control specifications.

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

Published online: 07 February 2012

Small items can often be assumed to be parallelepipeds, even if their actual shapes are not exactly so, but, generally, such approximation does not realistically hold when dealing with large items. Clusters of mutually orthogonal parallelepipeds, such as *tetris*-like items, or composite prisms are in most cases more suitable. Similar considerations hold, moreover, as far as the container shape is concerned.

The literature on the optimization of multidimensional packing problems (well known for being NP-hard) is extensive and advanced methods are available to solve difficult instances efficiently (e.g. [5,9,20,24]). Even if remarkable works concerning non-standard packing problems are available in the specialist literature (e.g. [18,27]) most of the research focuses on the orthogonal placement of rectangular items into rectangular domains, with no additional conditions. Non-standard packing problems with additional conditions have been tackled by dedicated heuristics or meta-heuristics, but possible approaches, based on mixed integer programming (MIP), have also been investigated (e.g. [2,7,8,25]).

When dealing with non-standard packing problems in the presence of overall conditions, such as balancing, the quite simplistic approach of placing items one after another results in being of very scarce efficiency and a global optimization (GO) point of view becomes a real necessity. The work reported here is addressed to the mixed integer linear/non-linear programming (MIP, MINLP) and exact GO framework (e.g. [11–14,16,17,19,21–23,26]).

Two classes of non-standard packing problems are considered here:

– packing of (three-dimensional) *tetris*-like items, with orthogonal rotations, within a convex domain (polyhedron)
– packing of convex and non-convex (simple) polygons with (continuous) rotations in a convex domain (polygon).

In both cases the objective function consists of maximizing the loaded volume or mass. Section 2.1 reports an MIP model to tackle the *tetris*-like item packing problem.
This model is described at a detailed level in the two previous works [7,8]. As the second part of this article refers to a *tetris*-like approximation of the polygon packing problem, the MIP model is reviewed with the scope of making the proposed approach self-explanatory. Similarly, Sect. 2.2 gives a high level description of a heuristic approach put forward in [8], where more details can be found. It has been adopted to efficiently solve the MIP model in practice. Some experimental details are reported.

Whilst the *tetris*-like item problem is very challenging, packing issues involving both convex and non-convex polygons are even more complex. In particular, while the non-intersection (necessary and sufficient) conditions are quite straightforward for orthogonal packing problems involving *tetris*-like items, the issue becomes much more complicated when polygons have to be dealt with.

Different approaches to this class of problems have been investigated (e.g. [4]). Fischetti and Luzzi [10] proposed an MIP approach for modelling the placement of a given set of pre-oriented (simple) polygons into a rectangular domain whose length has to be minimized. The adopted approach exploits the concept of no-fit and containment polygons.

Stoyan et al. [27] have introduced the Φ-functions concept for modelling complex two-dimensional packing problems. The placement of a given set of non-convex polygons, with the possibility of (continuous) rotations, into a strip, minimizing its length, is considered. The problem is formulated in terms of mathematical programming. An initial feasible solution is looked into by the sequential placing of polygons approximated by clusters of rectangles with prefixed (orthogonal) orientation. A local optimization approach is then performed by perturbing item positions and rotations.

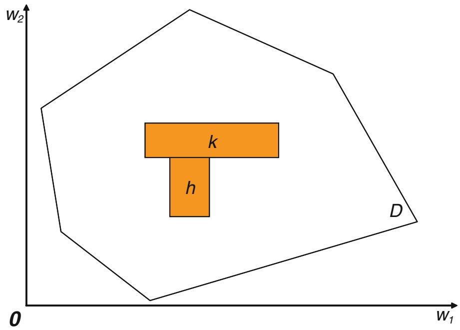


Fig. 1 *Tetris*-like item (two-dimensional representation)

A GO point of view, based on a successive approximation procedure, is instead presented here. Section 3.1 is devoted to an MINLP approach aimed at tackling the polygon problem (possible generalizations including 3-dimensional polyhedrons are not considered, putting them off to future research). Section 3.2 points out that a *tetris*-like formulation can be profitably adopted to look into a good initial (approximate) solution to the MINLP process; possible extensions are further outlined. Stoyan's Φ-functions could be adopted to refine the obtained solutions, in particular when very complex item shapes are involved.

2 *Tetris*-like items packing

2.1 MIP formulation

This section briefly reviews the *tetris*-like item problem MIP formulation described at a detailed level in [8]. A *tetris*-like item (Fig. 1), hereinafter simply referred to as *item*, when no ambiguity can occur, consists of a cluster of mutually orthogonal parallelepipeds (h and k in the figure), each one representing a component. In the following, the set of components associated to the generic item i is denoted by C_i. The problem under consideration is stated as follows.

Given a set of n items and a convex domain D, consisting of a polyhedron or approximated as such, place items into D maximizing the loaded volume (or mass), with the following positioning rules, for all picked items:

- for each item, each component side has to be parallel to an axis of a prefixed orthonormal reference frame (orthogonality conditions)
- for each item, each component has to be contained within D (domain conditions)
- components (of different items) cannot overlap (non-intersection conditions).

An orthogonal reference frame with origin O and axes $w_{\beta,}$ $\beta \in \{1, 2, 3\}$, is defined, while a local reference frame is associated to each item. Each local reference frame is chosen so that all item components lie within its first (*positive*) octant. Its origin coordinate, with respect to the main reference frame w_β axes, is denoted in the following by $o_{\beta i}$. We shall then introduce the set Ω of all possible orthogonal rotations, admissible for any item local reference frame, with respect to the main one. It is easily seen they are twentyfour in all, since items are in general asymmetric objects.

This is illustrated by Fig. 2, where an item, consisting of three mutually orthogonal components, is considered. The components have lengths of 1, 3 and 9 units, respectively. The component of length 3 units is parallel to the vertical axis of the observer reference frame.

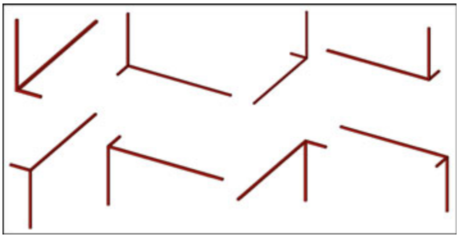


Fig. 2 *Tetris*-like item rotations around an axis

Two sub-cases are considered: in one (corresponding to the four images above) the item is *up-oriented*, while in the other (corresponding to the four images below) it is *down-oriented*. As can be seen from the figure, four orthogonal (clock-wise) rotations (around the vertical axis) are associated to each sub-case, so that, when the component of length 3 units is vertical (either *up-oriented* or *down-oriented*) eight relative rotations have to be taken into account. The same holds when either the component of length 1 unit or the one of length 9 units assume the vertical position, so that the total number of orthogonal orientation is twenty four.

We shall introduce for each item i the set E_{hi} of all (8) vertices associated to each of its component h. An extension of this set is obtained by adding to E_{hi} the geometrical centre of component h. The so defined extended set is denoted in the following by E^*_{hi}. For each item i and each possible (orthogonal) orientation ω we define the following binary variables:
$\chi_i \in \{0,1\}$, with $\chi_i = 1$ if item i is picked, $\chi_i = 0$ otherwise;

$\vartheta_{\omega i} \in \{0, 1\}$, with $\vartheta_{\omega i} = 1$ if item i(is picked and it) has the (orthogonal) orientation $\omega \in \Omega$, $\vartheta_{\omega i} = 0$ otherwise.

The above *orthogonality* conditions can be expressed as follows:

$$\forall i \quad \sum_{\omega} \vartheta_{\omega i} = \chi_i, \tag{1}$$

$$\forall \beta, \forall i, \forall h \in C_i, \forall \eta \in E^*_{hi} \quad w_{\beta\eta hi} = o_{\beta i} + \sum_{\omega} W_{\omega\beta\eta hi} \vartheta_{\omega i}, \tag{2}$$

where:
$w_{\beta\eta hi}$ $\left(\forall \eta \in E^*_{hi}\right)$ are the vertex coordinates of component h, or its geometrical centre ($\eta = 0$), relative to item i, with respect to the reference frame axes w_β;
$W_{\omega\beta\eta hi}$ are the coordinate differences between points $\forall \eta \in E^*_{hi}$ and the origin of the local reference frame, projected along the w_β axis of the main reference frame, corresponding to the (item i) orientation ω.
The *domain* conditions are expressed as follows:

$$\forall \beta, \forall i, \forall h \in C_i, \forall \eta \in E_{hi} \quad w_{\beta\eta hi} = \sum_{\gamma} V_{\beta\gamma} \lambda_{\gamma\eta hi}, \tag{3}$$

$$\forall i, \forall h \in C_i, \forall \eta \in E_{hi} \quad \sum_{\gamma} \lambda_{\gamma\eta hi} = \chi_i, \tag{4}$$

where:
$w_{\beta\eta hi} (\eta \in E_{hi})$ are the vertex coordinates of component h relative to item i, with respect to the reference frame axes w_β;
$\lambda_{\gamma\eta hi}$ are non-negative variables;
$(V_{11}, V_{21}, V_{31}), \ldots, (V_{1u}, V_{2u}, V_{3u})$ are the vertices of D (whose coordinates, in the main reference frame, are assumed as non-negative, with no loss of generality);

The *non-intersection* conditions are represented by the constraints below:

$$\forall\beta,\forall i,\forall j, i<j,\forall h\in C_i,\forall k\in C_j \\ w_{\beta 0hi}-w_{\beta 0kj}\geq \tfrac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i}+L_{\omega\beta kj}\vartheta_{\omega j}\right)-D_\beta\left(1-\sigma^{+}_{\beta hkij}\right), \tag{5-1}$$

$$\forall\beta,\forall i,\forall j, i<j,\forall h\in C_i,\forall k\in C_j \\ w_{\beta 0kj}-w_{\beta 0hi}\geq \tfrac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i}+L_{\omega\beta kj}\vartheta_{\omega j}\right)-D_\beta\left(1-\sigma^{-}_{\beta hkij}\right), \tag{5-2}$$

$$\forall\beta,\forall i,\forall j, i<j,\forall h\in C_i,\forall k\in C_j \\ \sum_{\beta}\left(\sigma^{+}_{\beta hkij}+\sigma^{-}_{\beta hkij}\right)\geq \chi_i+\chi_j-1, \tag{6-1}$$

$$\forall\beta,\forall i,\forall j, i<j,\forall h\in C_i,\forall k\in C_j \\ \sum_{\beta}\left(\sigma^{+}_{\beta hkij}+\sigma^{-}_{\beta hkij}\right)\leq \chi_i, \tag{6-2}$$

$$\forall\beta,\forall i,\forall j, i<j,\forall h\in C_i,\forall k\in C_j \\ \sum_{\beta}\left(\sigma^{+}_{\beta hkij}+\sigma^{-}_{\beta hkij}\right)\leq \chi_j, \tag{6-3}$$

where D_β are the sides of the parallelepiped of minimum volume enveloping D; $w_{\beta 0hi}$ and $w_{\beta 0kj}$ are the centre coordinates of components h and k; $L_{\omega\beta hi}$ and $L_{\omega\beta kj}$ are their sides, parallel to the w_β axis, corresponding to the orientation ω; $\sigma^{+}_{\beta hkij}\in\{0,1\}$ and $\sigma^{-}_{\beta hkij}\in\{0,1\}$ (it should be noticed that both constraints (6-2) and (6-3) are not necessary conditions and just play a role in tightening the MIP model). If in (5) a σ variable is set to one, the corresponding constraint is denoted by *relative position* constraint.

The constants D_β appearing in constraints (5-1) and (5-2) could be substituted with any constants $M_{\beta hkij}$ arbitrarily big (*big-M*), up to making the relative constraint redundant when the corresponding σ is zero. The following proposition holds.

Proposition 1 *If the domain D is a parallelepiped, constants $M_{\beta hkij}$ cannot be inferior to the corresponding D_β.*

Proof Consider any two items i, j and any relative components h and k, respectively and write constraints (5-1) and (5-2) in the form:

$$\forall\beta,\forall i,\forall j, i<j,\forall h\in C_i,\forall k\in C_j \\ w_{\beta 0hi}-w_{\beta 0kj}\geq \tfrac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i}+L_{\omega\beta kj}\vartheta_{\omega j}\right)-M_{\beta hkij}\left(1-\sigma^{+}_{\beta hkij}\right), \\ \forall\beta,\forall i,\forall j, i<j,\forall h\in C_i,\forall k\in C_j \\ w_{\beta 0kj}-w_{\beta 0hi}\geq \tfrac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i}+L_{\omega\beta kj}\vartheta_{\omega j}\right)-M_{\beta hkij}\left(1-\sigma^{-}_{\beta hkij}\right).$$

If $\sigma^{+}_{\beta hkij}=1$, the non-intersection condition between h and k is attained with respect to the corresponding β axis. In this case the coefficient of $M_{\beta hkij}$ becomes zero and the relative *big-M* is inactive. The same holds if $\sigma^{-}_{\beta hkij}=1$. If, instead, $\sigma^{+}_{\beta hkij}=0$, the inequality below must hold for any possible position of i and j in D:

$$M_{\beta hkij}\geq -w_{\beta 0ki}+w_{\beta 0hj}+\frac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i}+L_{\omega\beta kj}\vartheta_{\omega j}\right).$$

To prove the statement it is then sufficient to consider the limit cases, where item i and j enveloping rectangles are, with respect to the D_β side, at its opposite limit positions (i.e. at their maximum relative distance). Suppose, for instance, that item i enveloping rectangle, is at the D_β side upper limit and item j enveloping rectangle at the lower one. Denoting, then, for any rotation ω and ω' of i and j, respectively, by $L_{\omega\beta i}$ and $L_{\omega'\beta j}$ the side projection of their enveloping rectangles on $D_{\beta,}$ it is seen by simple computations that the following inequality must hold:

$$M_{\beta hkij} \geq D_\beta - L_{\omega\beta j} - L_{\omega'\beta j} + L_{\omega\beta hi} + L_{\omega'\beta hj}.$$

As it must hold for any ω and ω', it is true also when $-L_{\omega\beta j} - L_{\omega'\beta j} + L_{\omega\beta hi} + L_{\omega'\beta hj} = 0$. The same reasoning occurs obviously if $\sigma^-_{\beta hkij} = 0$, so that $M_{\beta hkij} \geq D_\beta$. □

As pointed out in [8], a number of additional conditions, such as the pre-fixed position/orientation of some items, a minimum gap between objects, forbidden zones (*holes*) inside the domain or the presence of separation planes (movable within given ranges), as well as the statement that the overall centre of mass must stay within a given (convex) domain (*balancing* conditions) can be easily taken into account, by extending the basic model reported above.

A particular version of this model can be formulated when the problem is expressed in terms of a feasibility one, that is when no objective function is a priori posed and all the given items have to be picked. This issue arises in practice, when, for instance, all the elements of a certain system have to be accommodated into a given domain and no one may be excluded. Concerning the context considered herewith, the feasibility problem formulation becomes of particular interest when dealing with the heuristic procedure described in the next section. Since no objective function is specified, any arbitrary one can be profitably introduced in order to make easier the task of looking into an integer-feasible solution.

It is well known that a major difficulty in MIP problem modelling and solving concerns the issue of tightening bounds. The work [28] deals with fixed charge (large-scale) models and presents an efficient pre-processing method aimed at minimizing the *big-M* terms, that is, at reducing (a priori) the region delimited by the corresponding fixed charge constraints in the LP-relaxation. An approach, aimed at minimizing the search region R, relative to the *non-intersection* (*big-M*) constraints (5), is described hereinafter to efficiently tackle the above feasibility problem. Constraints (5) are reformulated in an LP-relaxed form and an ad hoc objective function is introduced.

The reformulated model is reported below. All variables χ_i are set to one, as all the given items must be inside the domain. The *non-intersection* constraints are rewritten as:

$$\begin{aligned}&\forall\beta, \forall i, \forall j, i < j, \forall h \in C_i, \forall k \in C_j\\ &w_{\beta 0hi} - w_{\beta 0kj} \geq \tfrac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i} + L_{\omega\beta kj}\vartheta_{\omega j}\right) + d^+_{\beta hkij} - D_\beta,\end{aligned} \tag{7-1}$$

$$\begin{aligned}&\forall\beta, \forall i, \forall j, i < j, \forall h \in C_i, \forall k \in C_j\\ &w_{\beta 0kj} - w_{\beta 0hi} \geq \tfrac{1}{2}\sum_{\omega}\left(L_{\omega\beta hi}\vartheta_{\omega i} + L_{\omega\beta kj}\vartheta_{\omega j}\right) + d^-_{\beta hkij} - D_\beta.\end{aligned} \tag{7-2}$$

Constraints (6) are substituted with the following:

$$\forall\beta, \forall i, \forall j, i < j, \forall h \in C_i, \forall k \in C_j \qquad d^+_{\beta hkij} \geq \sigma^+_{\beta hkij} D_\beta, \tag{8-1}$$

$$\forall\beta, \forall i, \forall j, i < j, \forall h \in C_i, \forall k \in C_j \qquad d^-_{\beta hkij} \geq \sigma^-_{\beta hkij} D_\beta, \tag{8-2}$$

$$\forall i, \forall j, i < j, \forall h \in C_i, \forall k \in C_j \qquad \sum_{\beta=1}^{3}\left(\sigma^+_{\beta hkij} + \sigma^-_{\beta hkij}\right) = 1, \tag{9}$$

where $d^+_{\beta hkij}$ and $d^-_{\beta hkij} \in [0, D_\beta]$. The adopted ad hoc objective function is:

$$\max \sum_{\beta,h,k,i<j} (d^+_{\beta hkij} + d^-_{\beta hkij}). \tag{10}$$

The conditions $\forall \beta, \forall i, \forall j, i < j, \forall h \in C_i, \forall k \in C_j \; d^+_{\beta hkij} + d^-_{\beta hkij} \leq D_\beta$ could be advantageously added to tighten the feasibility region, without excluding any solution. Any optimal solution of the reformulated model identifies a minimal subset of the region R, relative to the original model. When an LP-relaxation is performed, by dropping the integrality conditions on the σ variables, the objective function (10), tends to induce the d variables to attain their upper bounds, so that, indirectly, it *minimizes* the overall overlapping of items. This aspect is advantageously exploited by the heuristic procedure described in the next section. When the problem is expressed in terms of feasibility (i.e. the set of items to load is fixed a priori) the following proposition holds.

Proposition 2 *For any fixed set of items, the solution regions, associated to the basic and the reformulated model respectively (apart from variables d^+ and d^-), are coincident.*

Proof Dealing with the feasibility problem, all χ variables, corresponding to the selected set of items that have to be loaded are set to one. In such a way, constraints (1), (2), (3) and (4) are obviously coincident in both models and it is thus sufficient to point out that constraints (5) and (6) of the basic model are equivalent to constraints (7), (8) and (9) of the reformulated model. It is immediately seen that, being that all χ variables are set to one, constraints (6) are reduced to constraints (9). To show that constraints (5) are equivalent to (7) and (8) we shall distinguish the cases where the σ variables are zero from the one where they are equal to one. For instance, consider $\sigma^+_{\beta hkij} = 0$. This implies that constraints (5-1) become:

$$w_{\beta 0hi} - w_{\beta 0kj} \geq \frac{1}{2} \sum_{\omega} \left(L_{\omega\beta hi} \vartheta_{\omega i} + L_{\omega\beta kj} \vartheta_{\omega j} \right) - D_\beta.$$

They are equivalent to constraints (7-1), with $d^+_{\beta hkij} = 0$, in compliance with constraints (8-1). Considering, instead, $\sigma^+_{\beta hkij} = 1$, this implies that constraints (5-1) become:

$$w_{\beta 0hi} - w_{\beta 0kj} \geq \frac{1}{2} \sum_{\omega} \left(L_{\omega\beta hi} \vartheta_{\omega i} + L_{\omega\beta kj} \vartheta_{\omega j} \right).$$

They are equivalent to constraints (7-1), with $d^+_{\beta hkij} = D_\beta$, in compliance with constraints (8-1). As the same reasoning can be carried out, taking into account the cases relative to the $\sigma^-_{\beta hkij}$ variables, the two models are equivalent.

2.2 Heuristic approach

Denoting by N the total number of couples (h, k) of components belonging to different items (from a set of n elements), any model instance contains:

$O(24n)$(binary) variables ϑ,
$O(n)$(binary) variables χ,
$O(6N)$(binary) variables σ,
O(7N) non-intersection constraints.

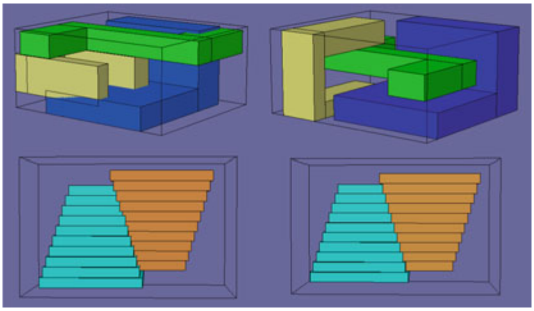


Fig. 3 Approximate (*left*) and exact (*right*) solutions

When dealing with real world instances, the resulting problems are very hard to solve, in particular because of the presence of a very large number of *big-M* constraints, essentially related to the *non-intersection* conditions (e.g. an instance involving 50 items of 5 components each, gives rise to about 3×10^4 *non-intersection* constraints). A heuristic procedure has previously been introduced [8] to efficiently solve the MIP model, when each item consists of a single parallelepiped (i.e. *tetris*-like items with just a component). Its generalization, in order to deal with *tetris*-like items, is straightforward, acting on each single component. The proposed heuristic is a recursive procedure, based on the concept of *abstract configuration*. Given N couples of components (belonging to different items), an *abstract configuration* consists of N *relative position* constraints (non-intersection constraints (5) with their relative σ variable set to one), one and only one for each couple, giving rise to a feasible solution in any unbounded domain (while the original domain D is, on the contrary, bounded).

The proposed procedure aims at generating a sequence of good*abstract configurations* and at solving, step by step, a reduced MIP model obtained by eliminating, from the original one, all the redundant *non-intersection* constraints (i.e. *non-intersection* constraints not contemplated by the current*abstract configuration*). At each step, items are rejected, if necessary, to make the current *abstract configuration* compatible with the given domain D. The heuristic overall logic is based on the following modules: *Initialization, Abstract Configuration Generation, Packing, Hole-filling, Item-exchange*.

The goal of the *Initialization* module is to obtain a good approximate initial solution. The feasibility model described in Sect. 2.1 is adopted and the integrality condition on the σ variables is dropped. In this case constraints (8-1), (8-2), (9) can be replaced by: $\sum_{\beta=1}^{3}\left(\frac{d^{+}_{\beta hkij}}{D_\beta} + \frac{d^{-}_{\beta hkij}}{D_\beta}\right) \geq 1$.

As mentioned above, the ad hoc objective function has the job of *minimizing* the intersection between the components of different items. This is showed by Fig. 3, where LP-relaxed solutions (on the left, with possible intersection between items) are compared with the corresponding MIP solutions (on the right, with no intersection between items).

The approximate solution so obtained is given as input to the *Abstract Configuration Generation* module. It aims at generating an *abstract configuration,* starting from any approximate solution obtained by the *Initialisation/Hole-filling* modules. A non-intersection constraint (5) is selected for each pair of components (belonging to different items). For non-intersecting components, the satisfied non-intersection constraints are considered and when more than one non-intersection constraint is satisfied for the same pair of components, the one corresponding to the maximum relative distance between the coordinates of the components is selected. For intersecting items, the non-intersection constraint corresponding to the minimum overlapping is chosen. The so generated *abstract configuration* is given as input to the *Packing* module.

The goal of this module is to look into a solution to the MIP model including just the selected *non-intersection* constraints corresponding to the generated *abstract configuration* (it is easy to prove that in these conditions the integrality request on the σ variables can be dropped tout court). If a satisfactory solution is found, it is taken as the 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 (a stopping rule can be posed).

The *Hole-filling* 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 and a number of non-picked items are pre-selected as candidates to cover non-covered nodes of the *grid*. The *Hole-filling* module is based on an ad hoc *allocation* MIP model [8]. The approximate solution, obtained by this module, is utilised to generate an *abstract configuration* that is given as input to the *Packing* module.

The *Item-exchange* module is aimed at performing a *non-blind* local search by perturbing the *abstract configuration* relative to the current *Packing* module solution (to tentatively give rise to an improved *abstract configuration*). Non-loaded items are exchanged (in the current *abstract configuration*) with picked items, to increase the loaded volume (or mass). The approximate solution obtained by the *Item-exchange* module is adopted to generate a new *abstract configuration* for the *Packing* module.

The packing problems considered in this work, just because non-standard, are quite hard to classify and it is very difficult to perform statistics on them as well. The efficiency of the approach proposed depends on a variety of factors, since the complexity of the problems to solve (in addition to the number of items/components) are indeed strongly dependent on the characteristics of the items, components and domain involved, as well as on overall conditions such as the balancing ones. The separation planes, as it is even too obvious, significantly reduce the volume exploitation, especially when the items are quite different from each other, in terms of volumes, dimensions, and ratios between their dimensions. The difficulties related to the centre of mass domain tightness are however not independent from the item typologies (in terms of mass, volume, dimensions).

An experimental analysis (presently ongoing) has been carried out in the following environments:

XP Professional Service Pack 2, Core 2 Duo P8600, 2.40 GHz, 1.93 GB RAM;
CLP COIN-OR LP 2.0, as simplex solver;
CBC COIN-OR 2.0 as LP-based branch-and-cut library;
CGL COIN-OR 2.0, as library of cutting-plane generators [3];
IBM ILOG CPLEX Optimizer 12.3 (with preliminary results only).

Roughly speaking, it could be said that balancing conditions with some percentage of admissible off-centering (with respect to the container dimensions) can decrease the exploited volume by 15–20% and increase the computational effort by up 25–30%. These estimates are however very imprecise and indicate just a general trend. The results obtained so far (COIN-OR) showed that for instances involving 75–100 components, the *Initialization* phase can require 50–120 CPU s; for each execution step the *Abstract Configuration Generation* generally lasts <2 s, the *Packing* one 240–360 s; the *Hole Filling* and the *Item Exchange* phases, involving 25–35 non-loaded items, can require <10 and 2 s. respectively. Table 1 reports some case studies showing the overall CPU time requested to obtain satisfactory (non-proven optimal) solutions. All models utilized by the heuristic have been generated by

Table 1 Case studies overall CPU time consumption (s)

	Tot. number of items	Tot. number of components	Presence of structural elements/sep. planes/min. gap between items	Presence of balancing conditions	Occupied volume %	COIN-OR CPU time (s)	CPLEX CPU time (s)
Case study 1	3	8	No	No	~ 77	~540	~9
Case study 2	5	10	No	No	~70	~495	~8
Case study 3	6	14	No	No	~57	~585	~9
Case study 4	6	16	No	No	~50	~650	~17
Case study 5	3	31	Yes	Yes	~51	~7200	~44
Case study 6	3	8	Yes	No	~64	~87	~8
Case study 7	4	32	Yes	Yes	~50	~7200	~7190
Case study 8	2	20	Yes	Yes	~48	~133	~11
Case study 9	2	22	Yes	No	~57	~270	~18
Case study 10	4	23	Yes	Yes	~49	~587	~23

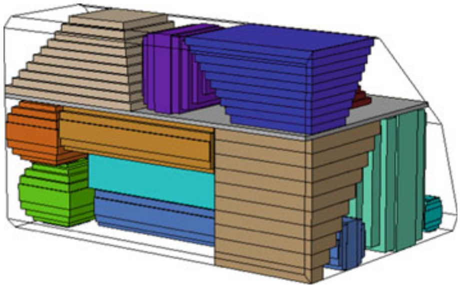


Fig. 4 *Tetris*-like items into a curved domain with a separation plane (~3 CPU hours, COIN-OR)

adopting the IBM MIP algebraic modeler [15]. The preliminary results obtained up to now by utilizing the IBM ILOG CPLEX Optimizer seem very promising (more than 80% reduction is in general expected), but further in-depth analysis is needed.

Figure 4 depicts a solution typology quite frequent in the framework of the space cargo accommodation, when curved containers are present (the curved shape offers the advantage of fitting the cylindrical domain of modules/vehicles, while the separation planes are generally adopted to facilitate the unloading operations).

A further example is reported here below, to give more insight into the experimental results, by pointing out both the input and output details. It consists of a *fabricated* instance, purposely built in order to deal with an easy-to-verify but nevertheless non-trivial problem, for which a high quality solution is known a priori. The proposed case study assumes thus the characteristic of a problem solving issue.

Given the main reference frame *(O, X, Y, Z)*, we shall consider a rectangular domain, with a vertex coincident with O and all the remaining ones within the first (positive) octant. The domain dimensions are $L_X = 13$, $L_Y = 11$ and $L_Z = 10$ (generic) units, respectively. The given set of items is reported in Table 2, having introduced, for each item i, the local reference frame $(O_i, x_i y_{i,} z_i)$. The first column enumerates the set of items, while the second one their components. Columns 3-4-5 report each component projection on the $x_{i,} y_{i,} z_i$ of the local reference frame axes respectively. Column 6 gives, for each component, its geometrical center coordinates, with respect to the relative item local reference frame (all data is expressed in generic units).

Table 3 reports, in the second column, the position obtained for each item local frame origin O_i, with respect to the main reference frame *(O,X,Y,Z)*. The remaining columns report all the non-zero direction cosines of the angles formed by the local and main reference frame axes respectively, for each item: xX, xY, xZ, yX, yY, yZ, zX, zY, zZ.

Two different views of the obtained solution are shown in Fig. 5, where components belonging to the same *tetris*-like item are denoted by the same relative item identifier. All the a-priori fabricated items of our instance have been properly loaded, attaining, as expected, 97.67% of the container volume. As can be gathered from Fig. 5, the *T-shaped* item (It1) is loaded upside down and partially covered by the surrounding items. The two C-shaped items (It2 and It3), in the figure, are rotated clock-wise. The total CPU time necessary to obtain Table 3 solution was about 110 s with COIN-OR and 40 s with CPLEX, respectively.

3 Polygons packing

3.1 MINLP formulation

Complex packing problems involving polygons have been considered in terms of MIP/MINLP (e.g. [10,18]).This section investigates some necessary conditions, formulated in terms

Table 2 Fabricated instance input

Items	Components	Component projection on x	Component projection on y	Component projection on z	Component centre coordinates (x, y, z)
It1 (*T*-shaped)	C11	3	4	6	5.5, 2, 3
	C21	11	4	4	5.5, 2, 8
It2 (*C*-shaped)	C11	4	4	8	2, 2, 4
	C21	5	4	4	6.5, 2, 2
	C31	4	4	8	11, 2, 4
It3 (*C*-shaped)	C13	4	4	8	2, 2, 4
	C23	5	4	4	6.5, 2, 2
	C33	4	4	8	11, 2, 4
It4 (single comp.)	–	2	4	13	1, 2, 6.5
It5 (single comp.)	–	2	4	13	1, 2, 6.5
It6 (single comp.)	–	3	5	6	1.5, 2.5, 3
It7 (single comp.)	–	3	4	10	1.5, 2, 5
It8 (single comp.)	–	3	4	5	1.5, 2, 2.5

Table 3 Fabricated instance output

Items	Local reference frame origin coordinates (X, Y, Z)	Local reference frame x axis orientation	Local reference frame y axis orientation	Local reference frame z axis orientation
It1 (*T*-shaped)	8, 11, 10	cos(xY) = −1	cos(yX) = −1	cos(zZ) = −1
It2 (*C*-shaped)	13, 0, 8	cos(xX) = −1	cos(yY) = 1	cos(zZ) = −1
It3 (*C*-shaped)	0, 11, 8	cos(xX) = 1	cos(yY) = −1	cos(zZ) = −1
It4 (single comp.)	0, 11, 8	cos(xZ) = 1	cos(yY) = −1	cos(zX) = 1
It5 (single comp.)	0, 4, 8	cos(xZ) = 1	cos(yY) = −1	cos(zX) = 1
It6 (single comp.)	8, 7, 0	cos(xY) = 1	cos(yX) = −1	cos(zZ) = 1
It7 (single comp.)	0, 7, 0	cos(xY) = 1	cos(yX) = −1	cos(zZ) = 1
It8 (single comp.)	8, 4, 6	cos(xZ)= 1	cos(yX) = 1	cos(zY) = 1

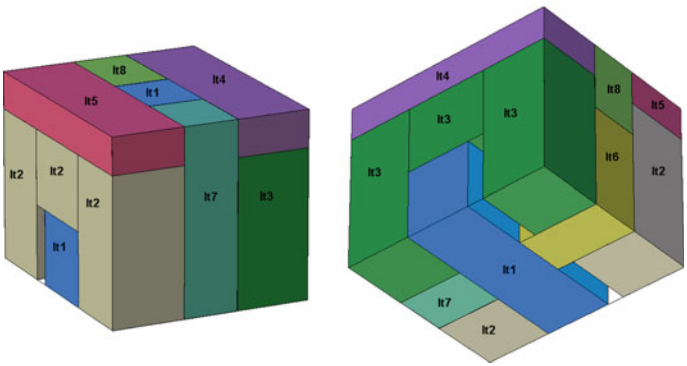


Fig. 5 Two views of the fabricated instance solution

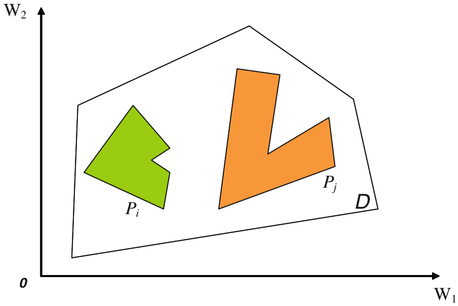


Fig. 6 Simple polygons inside a convex domain

of MINLP, to look into approximate solutions to the two-dimensional problem of placing (a subset out of n) simple polygons (i.e. polygons with no intersection between two non-consecutive edges) into a polygon D (convex domain), maximizing the overall surface of the loaded items (Fig. 6). For each polygon any possible orientation is admitted. A recursive process can be adopted to improve, by successive approximation, the so-far-obtained solution, until a satisfactory one is reached.

The problem positioning rules (for the picked items) are simply:

- each polygon has to be contained within D (domain conditions)
- polygons cannot overlap (non-intersection conditions)

To formulate the relative mathematical model, consider a given reference frame with origin O and axes w_β, $\beta \in \{1, 2\}$. The domain D consists of a polygon with u vertices. Their coordinates $(V_{11}, V_{21}), \ldots, (V_{1u}, V_{2u},)$, are assumed as non-negative (with no loss of generality) with respect to the (main) reference frame. We shall then consider any simple polygon i from the given set and associate to it a local reference frame $w_{\beta i}$, with origin O_i, of coordinates $o_{\beta i}$(with respect to the main reference frame). The set of all vertices associated to polygon i is denoted by E_i and the coordinates of any vertex $\eta \in E_i$ are denoted by $(V_{1\eta i,}\ V_{2\eta i})$. The vector equations below hold:

$$\forall i, \forall \eta \in E_i \qquad \boldsymbol{w_{\eta i}} = \chi_i \boldsymbol{o_i} + \chi_i \left\| q_{\beta\beta'} \right\|_i \boldsymbol{V_{\eta i}}, \tag{11}$$

where, for each vertex $\eta \in E_i$, $\boldsymbol{w}_{\rho i} = (w_{1\eta i}, w_{2\eta i})^T, \boldsymbol{o}_\mathrm{i} = (o_{1i}, o_{2i})^T, V_{\eta\mathrm{i}} = (V_{1\eta i}, V_{2\eta i})^T$, $\left\| q_{\beta\beta'} \right\|_\mathrm{i}$ is the (orthogonal) rotation matrix of the local reference frame, associated to polygon i, with respect to the main one ($\beta \in \{1, 2\}$) and $\chi_i \in \{0, 1\}$ has the same meaning assumed in Sect. 2.

The *domain* conditions below are stated to guarantee that each picked polygon i lay within the given polygon D:

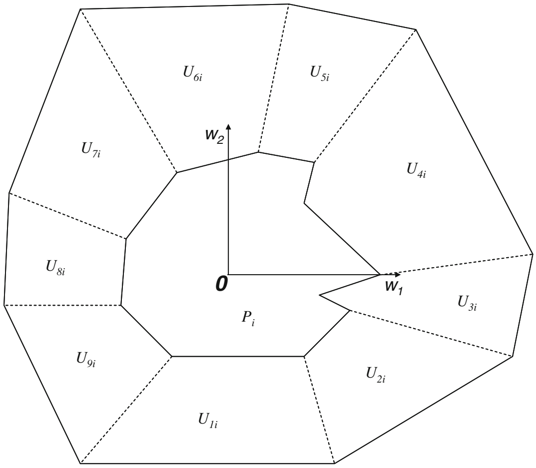
$$\forall \beta, \forall i, \forall \eta \in E_i \qquad w_{\beta\eta i} = \sum_{\gamma} V_{\beta\gamma} \lambda_{\gamma\eta i}, \tag{12}$$

$$\forall i, \forall \eta \in E_i \qquad \sum_{\gamma} \lambda_{\gamma\eta i} = \chi_i, \tag{13}$$

where $w_{\beta\eta i}$ are the vertex coordinates (defined by Eq. 11) of polygon i, the λ variables are non-negative and have the same meaning as in Sect. 2.

While in the case of *tetris*-like items the *non-intersection* conditions are quite easy to state, when dealing with polygons, they become much more complex. Three immediate-to-prove necessary conditions are posed here, to state a basis for the recursive process (acting by successive approximation). The following propositions are then stated.

Fig. 7 Augmented polygon



Proposition 3 *Given a set of internal circles S_i and S_j, for any pair of polygons i and j respectively, no circle of S_i can intersect a circle of S_j.*

Proposition 4 *For any pair of polygons i and j, no vertex of P_i can belong to any circle of S_j and vice versa.*

Proposition 5 *For each pair of polygons i and j, any set of points $\in P_i$ must belong to the external region of P_j and vice versa (this holds in particular for all vertices of the polygons).*

The third necessary*non-intersection* conditions (Prop. 5) posed above can be advantageously restricted to bounded external regions (*slices*, Fig. 7, see [10]). To this purpose the concept of *augmented polygon* is introduced, by posing the following definitions.

Definition 1 (*Augmented polygon)* For each polygon i consider the polygon $\overline{P}_i$ such that:

$$\begin{aligned} &P_i \subset \overline{P}_i \\ &\overline{P}_i - P_i = \bigcup_{\nu \in Q_i} U_{\nu i} \end{aligned}$$

where $U_{\nu i,}$ are convex polygons (not necessarily disjoint) associated to polygon i and Q_i is their set. Each $\overline{P}_i$, so obtained, is called augmented polygon associated to polygon i. (Notice that $\overline{P}_i - P_i$ could always be partitioned into a set of triangles).

Definition 2 (*Domain-covering augmented polygon)* For each polygon i, any associated augmented polygon that covers the whole domain D, for any position and orientation of i within D, is called domain-covering augmented polygon associated to polygon i and it is denoted by $\overline{\overline{P}}_i$.

The third necessary *non-intersection* conditions (Prop. 5), restricted to bounded external regions, can then be expressed as follows (Fig. 8):

for each pair of polygons i and j, with any associated $\overline{\overline{P}}_i$ and $\overline{\overline{P}}_j$, each point of P_i *must belong to* $\overline{\overline{P}}_j - P_j$ and vice versa.

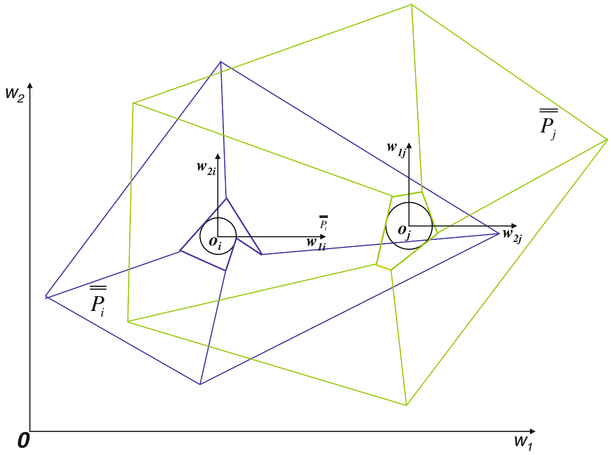


Fig. 8 Necessary *non-intersection* conditions

We shall now introduce, for each pair i, j of polygons the (implicitly binary) variable $\chi_{ij} \in [0, 1]$, with the following conditions:

$$\forall i, \forall j, i < j \qquad \chi_{ij} \le \chi_i, \tag{14-1}$$

$$\forall i, \forall j, i < j \qquad \chi_{ij} \le \chi_j, \tag{14-2}$$

$$\forall i, \forall j, i < j \qquad \chi_{ij} \ge \chi_j + \chi_j - 1. \tag{15}$$

The first necessary*non-intersection* conditions (Prop. 3) are expressed by the following constraints:

$$\forall i, \forall j, i < j, \forall h \in S_i, \forall k \in S_j, \\ \sum_{\beta} (o_{\beta hi} - o_{\beta kj})^2 \ge \chi_{ij}(R_{hi} + R_{kj})^2, \tag{16}$$

$$\forall i, \forall h \in S_i \qquad \boldsymbol{o}_{hi} = \chi_i \boldsymbol{o}_i + \chi_i \left\| \mathrm{q}_{\beta\beta'} \right\|_i \boldsymbol{O}_{hi}, \tag{17}$$

where S_i and S_j denote the (arbitrary) sets of internal circles associated to polygons i and j, R_{hi} and R_{kj} the radius of circles h and k respectively, $o_{\beta hi}$ and $o_{\beta ki}$ their centre coordinates, with respect to the reference frame. Vector equations 17 (as Eqs. 11) represent (with obvious meaning of the symbols), for the centre of circle h, the coordinate transformation between the local reference frame (associated to polygon i) and the main one. Prop. 4 formulation is very similar (and it is not reported hereinafter).

Once a *domain-covering augmented polygon* has been associated to each polygon i, the following constraints express the third necessary*non-intersection* conditions (Prop. 5):

$$\forall \beta, \forall i, \forall j, \forall \eta \in E_i \qquad \chi_{ij} w_{\beta\eta i} = \sum_{\substack{\gamma \in U_{\nu j} \\ \nu \in Q_j}} \mu_{\eta i \gamma \nu j} w_{\beta \gamma \nu j}, \tag{18}$$

$$\forall i, \forall j, \forall \eta \in E_i, \forall \nu \in Q_j \qquad \sum_{\gamma \in U_{\nu j}} \mu_{\eta i \gamma \nu j} = \delta_{\eta i \nu j}, \tag{19}$$

$$\forall i, \forall j, \forall \eta \in E_i \qquad \sum_{\nu \in Q_j} \delta_{\eta i \nu j} = \chi_{ij}, \tag{20}$$

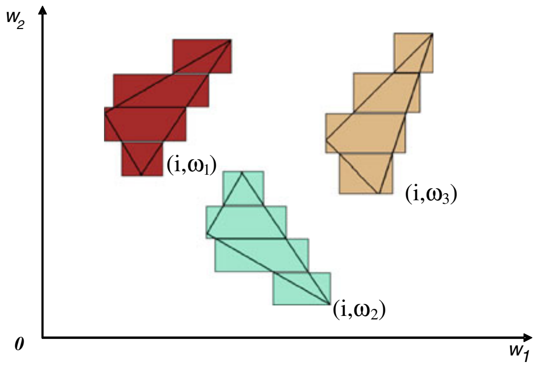


Fig. 9 Covering *tetris*-like items (corresponding to rotations $\omega_1, \omega_2, \omega_3$, of the same polygon i)

where $w_{\beta\eta i}$ are the coordinates of polygon i vertices (given by Eqs. 11) with respect to the main reference frame, and, similarly, $w_{\beta\gamma\nu j}$ are the coordinates of the $U_{\nu j}$ vertices, determining the *(domain-covering) augmented polygon j*; $\mu_{\eta i\gamma\nu j}$ are non-negative variables, $\delta_{\eta i\nu j} \in \{0, 1\}$ are binary variables. Constraints (19–20) ensure that, if both polygons i and j are loaded, each vertex of polygon i will belong to one (and only one) $U_{\nu j}$ and vice versa.

As the presence of the $\delta_{\eta i\nu j}$ binary variables increases the model complexity dramatically, constraints (19–20) could be profitably substituted by the following:

$$\forall i, \forall j, \forall \eta \in E_i, \forall \nu \in Q_j \qquad \sum_{\gamma \in U_{\nu j}} \mu_{\eta i\gamma\nu j} = \chi_{ij}. \tag{21}$$

In such a case, the logical restriction expressed by constraints (19–20) could then be treated algorithmically, by introducing proper special ordered sets, similar to the ones suggested by Escudero [6]. In particular they should manage the $\mu_{\eta i\gamma\nu j}$ variables so that only the ones corresponding to a single $U_{\nu j} \in Q_j$ are allowed to be positive, while all the remaining are forced to be zero.

3.2 Application and extensions

As in the case of the *tetris*-like items, because of the complexity of the problem per se, even when non-large-scale instances are involved, it is not expected to solve the polygon MINLP model tout court and a step-by-step procedure is thus strongly recommended. As a first rough approximation, the MINLP model solution process can be executed taking into account, for each polygon i, just one of the biggest internal circles (Fig. 8). The number of internal circles to take into account (Eq. 16–17) can be sequentially increased, for all pairs of polygons intersecting, until a satisfactory solution is attained.

As it is well known, however, when dealing with MINLP problems, a major difficulty in obtaining a satisfactory solution, in a reasonable computational time, strongly depends on the availability of a good initial guess. In the present case, this can be profitably looked into by temporary replacing polygons with covering *tertris*-like items and considering, for each polygon i, a set $\Omega = \{1, \ldots, p\}$ of possible (arbitrary) discretized rotations.

Given the generic polygon i, for each rotation $\omega \in \Omega$, define a *tetris*-like item (i,ω) (with sides orthogonal/parallel to the main reference frame axes), covering the polygon for that rotation (Fig. 9). The so defined *tetris*-like items are then characterized by the W'$_{w\beta hi}$ terms and by Eq. 2 of Sect. 2.1. In such a way, a single *tetris*-like item (i,ω) is associated to a single rotation ω of polygon i. *Tetris*-like items do not rotate any longer, but each of them correspond

Table 4 Polygon packing case study instance

Items	Vertex coordinates (local ref. frame)
It1: triangle	V_1(30,0), V_2(30,15)
It2: triangle	V_1(20,0), V_2(0,43)
It3: triangle	V_1(20,0), V_2(0,43)
It4: triangle	V_1(50,0), V_2(25,30)
It5: rectangle	V_1(30,0), V_2(30,75)
It6: rectangle	V_1(55,0), V_2(55,65)
It7: (*L*-shaped) irregular hexagon	V_1(45,0), V_2(43,11), V_3(5,11), V_4(10,39), V_5(−3,33)
It8: (*C*-shaped) irregular octagon	V_1(27,0), V_2(19,14), V_3(4,9), V_4(6,37), V_5(35,39),V_6(25,53), V_7(3,52)
It9: irregular dodecagon (quasi-rectan figure with two *handles*)	V_1(53,0), V_6(56,20), V_7(−1,20) (*)
It10: 16-sided figure (eight-pointed star)	V_1(7,0), V_2(10,−3), V_3(13,0), V_4(20,0) (*)
It11: 22-sided figure (jagged quasi-rectangular figure)	V_1(4,0), V_2(9,−3), V_3(10,0), V_{10}(55,0), V_{11}(52,17), V_{21}(2,17) (*)
It12: 32-sided figure (jagged quasi-squared figure)	V_1(5,−3), V_2(8,0), V_7(32,0), V_{16}(33,35), V_{23}(−2,35) (*)
It13: irregular quadrilateral	V_1(45,0), V_2(25,33), V_3(−2,22)
It14: irregular quadrilateral	V_1(21,0), V_2(31,5), V_3(−9,28)
It15: triangle	V_1(52,0), V_2(27,25)
It16: (*L*-shaped) irregular hexagon	V_1(45,0), V_2(43,11), V_3(5,11), V_4(10,39), V_5(−3,33)
It17: (*C*-shaped) irregular octagon	V_1(27,0), V_2(19,14), V_3(4,9), V_4(6,37), V_5(35,39),V_6(25,53), V_7(3,52)

(*) only a subset of vertices reported

to a possible (discretized) rotation ω of a polygon i. The MIP model formulation reported in Sect. 2.1 holds unaltered (provided that for each polygon i at most a single item (i,ω) is loaded) and can be directly adopted to obtain a good starting solution for the original MINLP model. As any *tetris*-like item covers the corresponding polygon for a specific rotation of it, any feasible solution of the *tetris*-like item problem is also a feasible solution of the polygon problem and represents a lower bound. Once a good initial solution is attained, the actual items (polygons) can be taken into account and the GO process carried out, by adding new items, until a satisfactory ultimate solution is obtained.

A heuristic approach, currently at a preliminary stage, performs the above *tetris*-like approximation as a first step. Then different overall packing techniques, such as item fixing, item-item exchange and *hole filling* are adopted by introducing, time after time, a number of the MINLP necessary conditions described in this section. In such a way the scale of the original instance can be dramatically reduced, significantly limiting the number of variables and constraints involved.

An illustrative case study is described hereinafter. Table 4 reports the given set of items to load inside a rectangular domain of dimension 150 and 170 (generic) units respectively. Each one is identified by its vertex coordinates, expressed (in generic units) with respect to its relative local reference frame. The local reference frame is centered in vertex V_0(0,0) and its first axis contains vertex V_1. All vertices are ordered on the basis of an anticlockwise sequence. Table 5 reports the numerical results obtained, pointing out both the items loaded and the ones rejected. The *tetris*-like step was up to loading items 1–12. This first step was carried out by

Table 5 Polygon packing case study numerical results

Items	V_0 coordinates (main ref. frame); α angle (rad.) between $V_1 - V_0$ and the X axes (main ref. frame)	Status
It1: triangle	$V_0(112{,}0)$; $\alpha = 0$	Loaded (*)
It2: triangle	$V_0(65{,}21)$; $\alpha = -2\pi/3$	Loaded (*)
It3: triangle	$V_0(52{,}170)$; $\alpha = -2\pi$	Loaded (*)
It4: triangle	$V_0(150{,}170)$; $\alpha = -2\pi$	Loaded (*)
It5: rectangle	$V_0(0{,}52)$; $\alpha = 0$	Loaded (*)
It6: rectangle	$V_0(93{,}72)$; $\alpha = 0$	Loaded (*)
It7: (L-shaped) irregular hexagon	$V_0(17{,}35)$; $\alpha = -0.87$	Loaded (*)
It8: (C-shaped) irregular octagon	$V_0(52, 130)$; $\alpha = -0.7$	Loaded (*)
It9: irregular dodecagon (quasi-rectan figure with two *handles*)	$V_0(120, 70)$; $\alpha = \pi/6$	Loaded (*)
It10: 16-sided figure (eight-pointed star)	$V_0(39{,}52)$; $\alpha = \pi/6$	Loaded (*)
It11: 22-sided figure (jagged quasi-rectangular figure)	$V_0(40{,}100)$; $\alpha = -\pi/6$	Loaded (*)
It12: 32-sided figure (jagged quasi-squared figure)	$V_0(70{,}73)$; $\alpha = -2.18$	Loaded (*)
It13: irregular quadrilateral	$V_0(3{,}135)$; $\alpha = -0.12$	Loaded (**)
It14: irregular quadrilateral	$V_0(2{,}12)$; $\alpha = -0.47$	Loaded (**)
It15: triangle	$V_0(130{,}34)$; $\alpha = 0.57$	Loaded (**)
It16: (L-shaped) irregular hexagon	–	Not loaded
It17: (C-shaped) irregular octagon	–	Not loaded

(*) Tetris-phase
(**) Hole-filling phase

concentrating unexploited areas in a limited number of empty zones, in order to make the task of a further *hole-filling* approach easier. This was achieved subsequently by fixing all items already loaded at their positions. Figure 10 shows on the left the results relative to the *tetris*-like step and on the right the obtained solution adopting LGO [23] as nonlinear optimizer (the whole process took about 40 min (COIN-OR was adopted as MIP solver for the *tetris*-like item approximation phase), but significant enhancements are expected improving the heuristic both in terms of solution efficiency and computational time. As the approach is currently under study and its relative analysis at a preliminary stage, an in-depth experimental activity is foreseen to estimate realistically the expected performances of the approach we are putting forward. Nonetheless, as far as the first phase is concerned, since it is based on the *tetris*-like approach described in Sect. 2, the experimental results reported there can give some sensible indication. Moreover, as Prop. 3 necessary conditions can be seen in terms of circle packing constraints, an insight concerning the relative computational effort can be found in [1].

The polygon packing problem described here is restricted to a two-dimensional case. The generalization to the three-dimensional problem is, in principle, straightforward. As for the *tetris*-like item case, balancing conditions could easily be added to the model. Possible extensions could also include non-simple polygons and non-convex domains. Some instances of non-simple polygons that can be treated with the proposed approach are illustrated by Fig. 11.

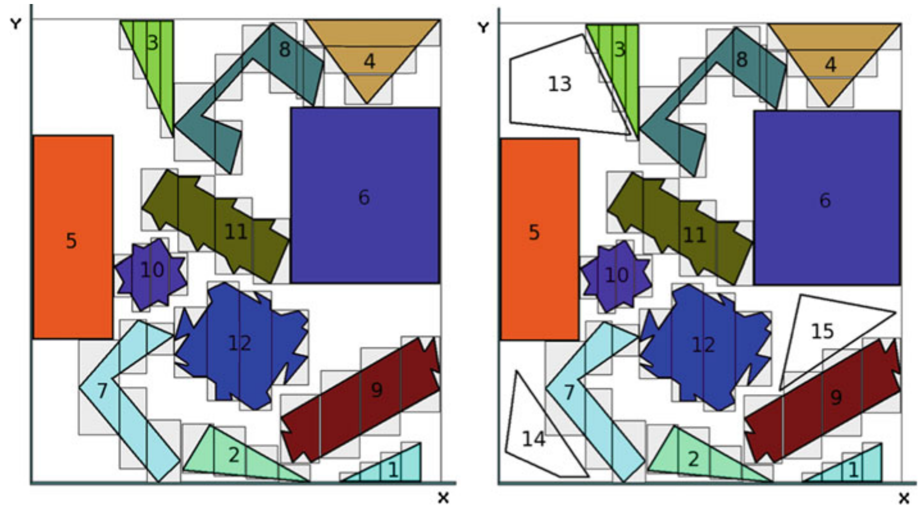
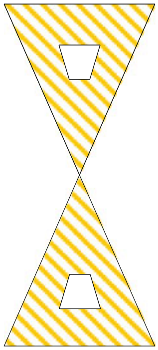


Fig. 10 Polygon packing case study graphical results

Fig. 11 Non-simple polygon (with holes)



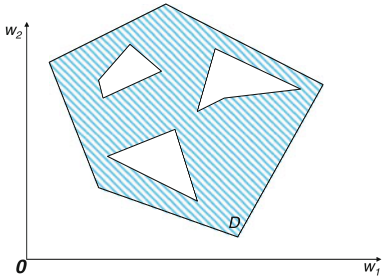
When patterns with *holes* are considered, the holes become part of the item external region. They have thus to be covered by convex polygons, to extend the item *domain-covering augmented polygon*. Non-convex domains can be easily modeled by fixing *virtual* (zero-mass) items that act as forbidden zones (Fig. 12).

The approach proposed in this section is aimed at finding satisfactory approximate global optimal solutions to the polygon packing problem. In some cases, however, a further refinement could be desirable. To this purpose the Stoyan's Φ-functions [28] could be profitably considered.

Given two (general two-dimensional) items $A_i(\boldsymbol{o}_i)$ and $A_j(\boldsymbol{o}_i)$, where $\boldsymbol{o}_i = (o_{1i}, o_{2i})$ and $\boldsymbol{o}_i = (o_{1ji}, o_{2ij})$ represent their local reference frame position respectively, any everywhere continuous function $\Phi_{ij}\!:\!\Re^4 \rightarrow \Re$ is called a Φ-function of $A_i(\boldsymbol{o}_i)$ and $A_j(\boldsymbol{o}_j)$ if it possesses the following characteristic properties:

$$\Phi_{ij} > 0 \text{ if } A_i(\boldsymbol{o}_i) \cap A_j\left(\boldsymbol{o}_j\right) = \emptyset;$$
$$\Phi_{ij} = 0 \text{ if } int\ A_i(\boldsymbol{o}_i) \cap int\ A_j\left(\boldsymbol{o}_j\right) = \emptyset \text{ and } \partial A_i(\boldsymbol{o}_i) \cap \partial A_j\left(\boldsymbol{o}_j\right) \neq \emptyset;$$
$$\Phi_{ij} < 0 \text{ if } int\ A_i(\boldsymbol{o}_i) \cap int\ A_j\left(\boldsymbol{o}_j\right) \neq \emptyset.$$

Fig. 12 Non-convex domain (with internal holes)



In such a way, $\Phi_{ij} \geq 0$ guarantees that items $A_i(\boldsymbol{o}_i)$ and $A_j(\boldsymbol{o}_j)$ don't intersect. An MINLP new model can thus be formulated by introducing a Φ-function for each couple of polygons. The approximated global solution obtained by the approach proposed in this article can be profitably utilized as an initial guess to efficiently solve the Φ-function-based MINLP model.

4 Conclusion and future work

The work presented in this paper originates from research carried out in space engineering, where very challenging non-standard packing problems with complex geometries, concerning both item and domain shapes, in the presence of additional conditions, such as balancing and operational requirements, are even more frequent. Similar applications can also arise in several different industrial fields relative to the transportation systems (e.g. aeronautical, naval, high-speed rail), as well as in logistics, manufacturing and system engineering.

The recent remarkable successes reached by the global optimization, in particular due to the approach based on exact methods, have given a boost to tackling very challenging packing problems from this point of view.

This paper presents two classes of non-standard packing problems. In both cases the loaded volume or mass has to be maximized. The first class concerns the packing of (three-dimensional) *tetris*-like items, with orthogonal rotations, within a convex domain (polyhedron). The second class concerns the packing of convex and non-convex simple polygons with (continuous) rotations in a convex domain (polygon), with possible additional conditions.

An MIP and MINLP formulations have been described respectively. The MIP model aimed at solving the *tetris*-like item problem has been discussed at a detailed level by the author in previous works and it has been briefly reviewed here, as its concept is preliminary to the second part of the article. The MINLP model aims at providing an (approximated) global optimal solution to the polygon problem. Both formulations are suitable to take into account non-trivial additional conditions, quite frequent in practice. A first (two-dimensional) *tetris*-like approximation is suggested to look into a good initial guess for the polygon packing problem and possible extensions to the MINLP formulation are outlined. In order to show possible applications of the approach proposed, a heuristic procedure, at present under study, is mentioned. In addition to its enhancement, future research has to be addressed to an in-depth experimental analysis and study of ad hoc MIP/MINLP strategies.

Acknowledgments The author is grateful to J.D. Pintér (PCS, Inc., Canada) for discussing the manuscript and his valuable suggestions concerning the GO and MINLP aspects considered in this work.

References

1. Castillo, I., Kampas, F.J., Pintér, J.D.: Solving circle packing problems by global optimization: numerical results and industrial applications. Eur. J. Oper. Res. **191**(3), 786–802 (2008)
2. Chen, C.S., Lee, S.M., Shen, Q.S.: An analytical model for the container loading problem. Eur. J. Oper. Res. **80**, 68–76 (1995)
3. COIN-OR Foundation, Inc.: COIN-OR Projects. http://www.coin-or.org/projects (2006)
4. Dowsland, K.A. et al.: Jostling for position: local improvement for irregular cutting patterns. J. Oper. Res. Soc. **49**, 647–658 (1998)
5. Dyckhoff, H., Scheithauer, G., Terno, J. et al.: Cutting and packing. In: Dell'Amico, M. (ed.) Annotated Bibliographies in Combinatorial Optimization, pp. 393–412. Wiley, Chichester (1997)
6. Escudero, L.: S3 sets. An extension of the Beale-Tomlin special ordered sets. Math. Program. **42**, 113–123 (1988)
7. Fasano, G.: A MIP approach for some practical packing problems: Balancing constraints and tetris-like items. 4OR: Q. J. Oper. Res. **2**(2), 161–174 (2004)
8. Fasano, G.: MIP-based heuristic for non-standard 3D-packing problems. 4OR: Q. J. Oper. Res. **6**(3), 291–310 (2008)
9. Fekete, S., Schepers, J.: A combinatorial characterization of higher-dimensional orthogonal packing. Math. Oper. Res. **29**, 353–368 (2004)
10. Fischetti, M., Luzzi, I.: Mixed-integer programming models for nesting problems. J. Heuristics **15**(3), 201–226 (2009)
11. Floudas, C.A., Pardalos, P.M.: A Collection of Test Problems for Constrained Global Optimization Algorithms. Springer, New York, NY, USA (1990)
12. Floudas, C.A., Pardalos, P.M. et al.: Handbook of test problems in local and global optimization. In: Noncovex Optimization and Its Applications, vol. 33. Kluwer, Dordrecht (1999)
13. Floudas, C.A., Pardalos, P.M. (eds.): Encyclopedia of Optimization. Kluwer, Dordrecht, The Netherlands ISBN-10: 0792369327 (2001)
14. Floudas, C.A. et al.: Global optimization in the 21st century: advances and challenges for problems with nonlinear dynamics. Comput. Chem. Eng. **29**, 1185–1202 (2005)
15. IBM Corporation: Easy Modeler User Guide SB135249-01 (1994)
16. Kallrath, J., Maindl, T.I.: Real Optimization with SAP-APO. Springer, Heidelberg (2006)
17. Kallrath, J.: Modeling difficult optimization problems. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, 2nd edn, pp. 2284–2297. Springer, New York (2008)
18. Kallrath, J.: Cutting circles and polygons from area-minimizing rectangles. J. Glob. Optim. **43**(2–3), 299–328 (2009)
19. Liberti, L., Maculan, N. (eds.): Global Optimization: From Theory to Implementation. Springer, New York (2005)
20. Martello, S., Pisinger, D., Vigo, D.: The three-dimensional bin packing problem. Oper. Res. **48**, 256–267 (2000)
21. Pardalos, P.M., Resende, M.G.H., (eds.): Handbook of Applied Optimization. Oxford University Press, USA, ISBN: 0195125940 (2002)
22. Pintér, J.D.: Mathematical programming glossary supplement: global optimization. Mathematical programming glossary, INFORMS computing society. http://glossary.computing.society.informs.org (2006)
23. Pintér, J.D.: Software development for global optimization. In: Pardalos, P.M., Coleman, T.F. Global Optimization: Methods and Applications, vol. 55, pp. 183–204, Fields Institute Communications, American Mathematical Society, Providence, RI (2009)
24. Pisinger, D., Sigurd, M.M.: The two-dimensional bin packing problem with variable bin sizes and costs. Discret. Optim. **2**(2), 154–167 (2005)
25. Pisinger, D., Sigurd, M.M.: Using decomposition techniques and constraints programming for solving the two-dimensional bin packing problem. INFORMS J. Comput. **19**(1), 36–51 (2006)
26. Rebennack, S. et al.: Column enumeration based decomposition techniques for a class of non-convex MINLP problems. J. Glob. Optim. **43**(2–3), 277–297 (2009)
27. Stoyan, Y.G. et al.: Φ-functions for complex 2D-objects. Q. J. Oper. Res. **2**(1), 69–84 (2008)
28. Suhl, U.H.: Solving large scale mixed integer programs with fixed charge variables. Math. Program. **32**(2), 165–182 (1984)