Chapter 6
Extensions and Mixed-Integer Nonlinear Approaches for Further Applications

The modelling approach advocated by this volume is susceptible to possible extensions. One, in particular, deals with the problem of looking into how the free volume of a container, partially loaded with *tetris*-like items, could be profitably exploited. *Virtual* items, i.e. (rectangular) parallelepipeds of non-prefixed dimensions, are purposely introduced to 'suggest' how to fill the empty volumes. In order to meet the practical demand, depending on the specific framework in question, limitations are stated on the maximum number of *virtual* items allowed, as well as on their minimum dimension. This issue is investigated hereinafter, highlighting a dedicated MIP formulation (Sect. 6.1).

The *global optimization* approach, stressed in this work, is further extended to the problem of packing simple polygons, with continuous rotations, inside a convex polygon. A heuristic approach, solving recursively a dedicated *mixed-integer nonlinear programming* (MINLP) model (founded on necessary conditions), is outlined (Sect. 6.2). It is aimed at providing an approximate global solution that can be further refined by exact local optimization-based methods. The *tetris*-like formulation is properly adapted to generate the first starting solution and profitably initialize the *mixed integer nonlinear programming* search process.

6.1 Exploiting Empty Volumes by Adding Virtual Items

This section is devoted to the issue of exploiting the residual volume of a container, partially loaded, by adding a certain number of *virtual* items. These are intended as (rectangular) parallelepipeds, not defined a priori in terms of dimensions. They are aimed at indicating how real items could still be loaded into the container.

This kind of problem arises, for instance, quite frequently in the framework of the logistic support to the International Space Station (ISS, cf. http://www.nasa.gov), when planning the periodical resupply of the resources stowed on board. A significant number of similar applications are expected in logistics in general. Here, indeed, the frequent necessity of introducing (rigid) packaging material, to

G. Fasano, *Solving Non-standard Packing Problems by Global Optimization and Heuristics*, SpringerBriefs in Optimization,
DOI 10.1007/978-3-319-05005-8_6, © Giorgio Fasano 2014

prevent item collisions, represents, from the analytical point of view, the same problem typology. Further examples, albeit in quite different fields, concern the use of autonomous robots (e.g. when requested to determine accessibility zones or to carry out assembling activities) and specific packing issues in the VLSI context.

The problem considered in this section can be formulated as follows:

Given a (convex three-dimensional) domain D, and a set of tetris-like items inside it, let us add a number (not exceeding the maximum value of $\overline{N}_V$*) of virtual items (i.e. rectangular parallelepipeds of variable dimensions). The total loaded volume is maximized by repositioning, if necessary, the (tetris-like) items already accommodated.*

Two classes of items are then taken into account: the tetris-like and *virtual*, that are single (rectangular) parallelepipeds, with no a-priori-given dimensions. All positioning rules of the general problem of Sect. 2.1, i.e. *orthogonality*, *domain* and *non-intersection* conditions, still hold. In particular:

- *Each virtual item has to be positioned orthogonally, with respect to the main reference frame.*
- *Each virtual item has to be contained within (the convex domain) D.*
- *Virtual items cannot overlap either with the tetris-like or other virtual ones.*

6.1.1 Model Formulation

An MINLP formulation of the optimization problem in question is considered first. We shall point out that the packing rules expressed above can be grouped as follows:

- *Orthogonality*, *domain* and *non-intersection* conditions for tetris-like items only
- *Orthogonality*, *domain* and *non-intersection* conditions for *virtual* items only
- *Non-intersection* conditions between tetris-like items and the *virtual* ones

As far as the first group is concerned, they are represented by constraints (2.1), (2.2), (2.3), (2.4), (2.5a), (2.5b) and (2.6) of Sect. 2.1. The *orthogonality*, *domain* and *non-intersection* conditions for *virtual* items only, as well as the *non-intersection* ones, between tetris-like items and the *virtual* ones, are quite straightforward. They are discussed here below.

Let us introduce the set of *virtual* items I_V, and the binary variables $\chi_{Vj} \in \{0, 1\}$, $j \in I_V$, with the meaning: $\chi_{Vj} = 1$ if *virtual* item j is included; $\chi_{Vj} = 0$ otherwise. For each *virtual* item j, $w_{V0\beta j}$ denote the centre coordinates, with respect to the main reference frame, and $l_{V\beta j}$ the side parallel to the axis w_β (of the main reference frame). E_{Vj} is the set of (eight) vertices associated to j whose coordinates are expressed as follows:

$$\begin{aligned}&\forall\beta\in B, \forall j\in I_V, \forall\eta\in E_{Vj}\\ &w_{V\beta\eta j} = w_{V\beta 0j} \mp \frac{1}{2} l_{V\beta j}.\end{aligned} \tag{6.1}$$

It is easily seen (cf. Sect. 2.1) that the *orthogonality* conditions are implicitly contemplated by equations (6.1), while the *domain* constraints have the same expressions of equations (2.3) and (2.4) (adopting the specific *virtual* item symbolism).

The following inequalities represent the *non-intersection* conditions between the generic tetris-like item i and the *virtual* one j, cf. constraints (2.5a), (2.5b) and (2.6):

$$\begin{aligned}&\forall\beta\in B, \forall i\in I, \forall j\in I_V, \forall h\in C_i\\ &w_{\beta 0hi} - w_{V\beta 0j} \geq \frac{1}{2}\sum_{\omega\in\Omega}\left(L_{\omega\beta hi}\vartheta_{\omega i}\right) + \frac{1}{2} l_{V\beta j} - D_\beta\left(1 - \sigma^+_{V\beta hij}\right),\end{aligned} \tag{6.2a}$$

$$\begin{aligned}&\forall\beta\in B, \forall i\in I, \forall j\in I_V, \forall h\in C_i\\ &w_{V\beta 0j} - w_{\beta 0hi} \geq \frac{1}{2}\sum_{\omega\in\Omega}\left(L_{\omega\beta hi}\vartheta_{\omega i}\right) + \frac{1}{2} l_{V\beta j} - D_\beta\left(1 - \sigma^-_{V\beta hij}\right),\end{aligned} \tag{6.2b}$$

$$\forall i\in I, \forall j\in I_V, \forall h\in C_i \quad \sum_{\beta\in B}\left(\sigma^+_{V\beta hij} + \sigma^-_{V\beta hij}\right) \geq \chi_i + \chi_{Vj} - 1, \tag{6.3}$$

$$\forall i\in I, \forall j\in I_V, \forall h\in C_i \quad \sum_{\beta\in B}\left(\sigma^+_{V\beta hij} + \sigma^-_{V\beta hij}\right) \leq \chi_i, \tag{6.4a}$$

$$\forall i\in I, \forall j\in I_V, \forall h\in C_i \quad \sum_{\beta\in B}\left(\sigma^+_{V\beta hij} + \sigma^-_{V\beta hij}\right) \leq \chi_{Vj}, \tag{6.4b}$$

where $\sigma^+_{V\beta hij}$ and $\sigma^-_{V\beta hij} \in \{0, 1\}$. The *non-intersection* constraints for *virtual* items only are immediately understood. The lower bound $\underline{L}_V$ is further introduced for all *virtual* item sides, in order to obtain acceptable solutions from a practical point of view (i.e. excluding ‘too small’ objects). The following constraints are thus stated:

$$\forall j\in I_V \quad \underline{L}_V\chi_{Vj} \leq l_{V\beta j} \leq D_\beta\chi_{Vj}. \tag{6.5}$$

Since the total volume of the *virtual* items added has to be maximized, the nonlinear *objective* function below is defined:

$$\max\sum_{j\in I_V}\prod_{\beta\in B} l_{V\beta j}. \tag{6.6}$$

Remark 6.1 It is gathered that the additional conditions discussed in Sect. 2.3 can easily be introduced (with proper adaptation, if necessary). As far as the static balancing ones (Sect. 2.3.4.1), in particular, are concerned, they have to be adequately extended to include the virtual items. To do this, we shall firstly assume that they have the (hypothetical average) density R_V. Expressions (2.23) and (2.24) are therefore modified as

$$\forall\beta\in B \quad \sum_{i\in I} M_i w^*_{\beta i} + R_V \sum_{j\in I_V} w_{V\beta 0j} \prod_{\beta\in B} l_{V\beta j} = \sum_{\gamma\in V^*} V^*_{\gamma\beta}\psi^*_{\gamma}, \quad \sum_{\gamma\in V^*} \psi^*_{\gamma} = m,$$

where $m = \sum_{i\in I} M_i + R_V \sum_{j\in I_V} \prod_{\beta\in B} l_{V\beta j}$, $\forall\ \gamma\ \in\ V^*$, $\psi^*_\gamma = \widetilde{\psi}^*_\gamma m$ and $\widetilde{\psi}^*_\gamma \geq 0$. As is easily seen, these conditions, differently from the case of Sect. 2.3.4.1, are no longer linear.

6.1.2 Model Approximations

A possible (quite daring) approximation of the MINLP model presented in Sect. 6.1.1 consists of adopting the following linear *objective* function as a *surrogate* of the nonlinear (6.6):

$$\max \sum_{\substack{\beta\in B/\\ j\in I_V}} l_{V\beta j}. \tag{6.7}$$

An alternative approach consists of replacing function (6.6) with a separable one and carrying out a piecewise linear approximation of each term (e.g. Williams 1993). This can easily be achieved by introducing the (likewise) *surrogate objective* function:

$$\max \sum_{\substack{\beta\in B,\\ j\in I_V/\\ l_{V\beta j}>0}} \ln\left(l_{V\beta j}\right). \tag{6.8}$$

This is indeed separable (and no longer a *surrogate* one, when just a single *virtual* item is considered, cf. Sect. 6.1.3). The piecewise linear approximation of each (single-variable) logarithmic term in (6.8) reduces then the original MINLP model to a much simpler (approximate) MIP one. A straightforward formulation is outlined here below (cf. Williams 1993 and Sect. 2.3.2).

For each axis w_β, we shall discretize the intervals $\left[\underline{L}_V, D_\beta\right]$ in a set $D_{SV\beta}$ of subintervals $[D_{V\beta(\gamma-1)}, D_{V\beta\gamma}]$ and then pose

$$\forall\beta\in B, \forall j\in I_V \quad l_{V\beta j} = \sum_{\gamma\in D_{SV\beta}} D_{V\beta\gamma}\lambda_{V\beta\gamma j}, \tag{6.9}$$

$$\forall \beta \in B, \forall j \in I_V \quad \ln\left(l_{V\beta j}\right) \approx \sum_{\gamma \in D_{SV\beta}} \ln\left(D_{V\beta\gamma}\right)\lambda_{V\beta\gamma j}, \tag{6.10}$$

$$\forall \beta \in B, \forall j \in I_V \quad \sum_{\gamma \in D_{SV\beta}} \lambda_{V\beta\gamma j} = \chi_{Vj}, \tag{6.11}$$

where the terms $\lambda_{V\beta\gamma j}$ are nonnegative variables.

It is worth noticing that, in this specific case (as in that of equations (2.20) of Sect. 2.3.2), the *adjacency* condition (for which *at most two adjacent λ can be non-zero*) may be dropped tout court (cf. Williams 1993), with significant computational benefit. It is, indeed, sufficient to observe that the optimization problem in question is equivalent to that of minimizing a convex *objective* function. This is immediately seen simply considering that expression (6.8) is equivalent to min $\sum_{\substack{\beta \in B, \\ j \in I_V}} \left[-\ln\left(l_{V\beta j}\right)\right]$ that is convex, as it is a sum (with positive coefficients) of convex functions (e.g. Minoux and Vajda 1986).

Both *objective* functions (6.7) and (6.8) are suitable for providing a starting approximated solution for the (exact) MINLP formulation of Sect. 6.1.1. A more refined (even if more demanding) approach could be followed as an alternative to avoid the introduction of *surrogate* functions. It is based on the method of converting products of (two or more) variables into *separable* functions, by means of quadratic terms (e.g. Williams 1993). It is briefly outlined here.

In the case of the product of two variables $q_1 q_2$, it is sufficient to introduce the new variables s_1 and s_2 (not restricted to be nonnegative), by performing the transformations $s_1 = \frac{1}{2}(q_1 + q_2)$, $s_2 = \frac{1}{2}(q_1 - q_2)$. The terms $q_1 q_2$ are hence substituted with $s_1^2 - s_2^2$ (that is a non-convex function). The method can be extended when the products involve more than two variables and a piecewise linear approximation of the quadratic terms can hence be achieved.

Remark 6.2 The task of minimizing the container area/volume can also be achieved by performing a logarithmic transformation and a piecewise linear approximation, without introducing any surrogate objective function (e.g. Pan and Liu 2006; Wang and Tsai 2010). The resulting model, nonetheless, appears quite complicated. Indeed, indicating with d_β the variables representing the container (parallelepiped) dimensions, with respect to the corresponding axes w_β, the objective function (in logarithmic form) $\sum_{\beta \in B} \ln\left(d_\beta\right)$ to minimize is not convex. As a consequence, the adjacency condition cannot be neglected.

Remark 6.3 The presence of the binary variables χ_{Vj} in expressions (6.5) guarantees that if a virtual item is not added, its contribution to the total volume is zero. This implication is nonetheless implicitly stated by equations (6.1) together with the domain constraints for virtual items.

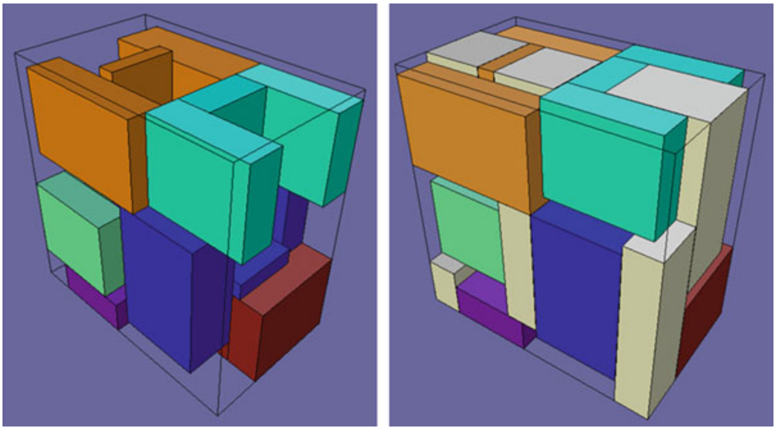


Fig. 6.1 *Virtual* item Case Study 6

6.1.3 Applications

A previous work (Fasano and Vola 2013) focuses on the utilization of the *surrogate* linear *objective* function (6.7), in the context of a dedicated heuristic approach. This has been conceived with the aim of obtaining quick satisfactory (but typically suboptimal) solutions to the original (nonlinear) problem. This approach is outlined, briefly, hereinafter, whilst the reader is referred to the work quoted above for more details, both on the algorithmic and experimental aspects.

An appropriate lower bound, as stated by conditions (6.5), is generated each time, depending on the specific instance to solve. The heuristic approach adopted assumes that an *abstract configuration*, relative to the already loaded tetris-like items, is provided. Since the addition of several *virtual* items, all together, would represent a significant computational effort, the heuristic procedure progresses incrementally. This is obtained by adding one *virtual* item after the other, until either a satisfactory solution is obtained or their maximum number is reached.

A currently ongoing experimental analysis (Fasano and Vola 2013) is being carried out. Some insights on the computational results, obtained to date, are briefly illustrated in Table A.11; see Appendix. They refer to a set of 32 case studies. For all tests considered, a maximum threshold of 10 *virtual* items was imposed, setting a runtime limit of 3 CPU hours. Case studies 6 and 22 are illustrated by Figs. 6.1 and 6.2, respectively (on the left the already loaded tetris-like items are shown and, on the right, the *virtual* ones added).

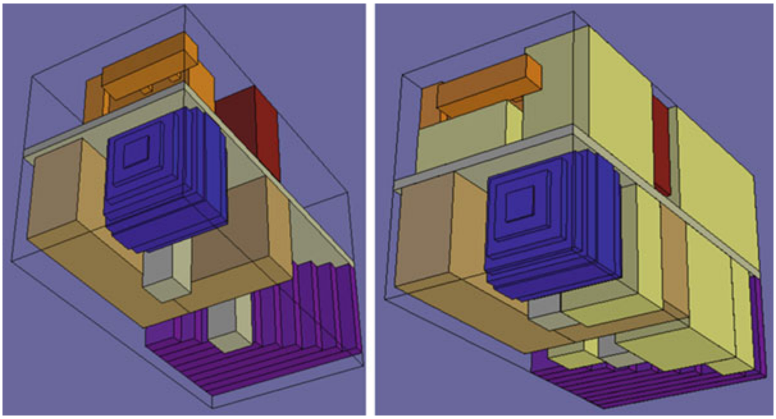


Fig. 6.2 *Virtual* item Case Study 22

6.1.3.1 Container Area/Volume Minimization by Maximizing Virtual Items

The concept of *virtual* item and the relative packing model can be utilized to solve the important issue of the container area/volume minimization. In its two-dimensional form, it consists of placing (orthogonally) a given set of single rectangles into one of minimum area (the generalization to the case of three-dimensional tetris-like items inside a parallelepiped is straightforward). We introduce first the domain D of sides D_β, $\beta = \{1, 2\}$ (the set of axes is still denoted by B, cf. Sect. 2.1), assumed sufficiently big to allow the loading of all items inside the corresponding rectangle. For all of them the constraints stated in the special case of Sect. 2.1 hold, with the corresponding variables χ set to one. Two *virtual* items are then introduced with the purpose of restricting the domain from the right and the upper edges, respectively. They are denoted as $\overline{R}$ and $\overline{U}$. $\overline{R}$ sides are $l_{1\overline{R}}$ (variable), parallel to D_1 and D_2 (constant). Its centre coordinates are indicated with $w_{\beta\overline{R}}$. The following specific (*domain*) constraints are posed:

$$w_{1\overline{R}} + \frac{1}{2} l_{1\overline{R}} = D_1, \tag{6.12a}$$

$$w_{2\overline{R}} = \frac{1}{2} D_2. \tag{6.12b}$$

Similar equations are set for the *virtual* item $\overline{U}$:

$$w_{1\overline{U}} = \frac{1}{2}\left(D_1 - l_{1\overline{R}}\right) \tag{6.13a}$$

$$w_{2\overline{U}} + \frac{1}{2} l_{2\overline{U}} = D_2 \tag{6.13b}$$

The *non-intersection* constraints (6.2a) and (6.2b) are specified as follows:

$$\forall i \in I \quad w_{1\overline{R}} - w_{1i} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 1 i}) + \frac{1}{2} l_{1\overline{R}}, \tag{6.14a}$$

$$\forall i \in I \quad w_{2\overline{U}} - w_{2i} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 2 i}) + \frac{1}{2} l_{2\overline{U}}, \tag{6.14b}$$

where the meaning of the symbols is understood (cf. Sect. 2.1, special case). It is immediate to see that, as a consequence, all the given items are confined to the inside of the rectangle cut out from D by $\overline{R}$ and $\overline{U}$. This rectangle has the vertex (0,0) in common with the domain D and the sides $D_1 - l_{1\overline{R}}$ and $D_2 - l_{2\overline{U}}$ laying on w_1 and w_2, respectively. The optimization problem under consideration consists hence of minimizing this area (obviously included within D) simply by adopting, for the *virtual* items $\overline{R}$ and $\overline{U}$, the *objective* function (6.6). This assumes, in the specific case, the following form (easily reducible to a *separable* function, even if non-convex, as outlined in Sect. 6.1.2):

$$\max\left(D_2 l_{1\overline{R}} - l_{1\overline{U}} l_{2\overline{U}}\right). \tag{6.15}$$

The heuristic approach presented in this section can thus be well adopted to obtain an approximate solution.

6.2 Non-orthogonal Packing of Non-rectangular Items

The key idea advocated in this monograph on non-standard packing problems, with additional conditions, also of ‘transversal’ nature (e.g. balancing), relies on a modelling-based GO approach. This point of view, espoused for the tetris-like item orthogonal packing, can be extended, at least at an approximate level, to more complex frameworks. This holds, in particular, for objects like polyhedrons, with the possibility of continuous rotations. According to the approach put forward here, the tetris-like formulation still plays an important role, providing a ‘naïve’ starting global solution.

The literature on the packing of complex (non-rectangular) objects is extensive (e.g. Bennell and Oliveira 2008; Betke and Henk 2000; Cagan et al. 2002; Chernov et al. 2010; Egeblad et al. 2009a, b; Gan et al. 2004; Kallrath 2009; Torquato and Jiao 2009), also including quite sophisticated formulations, but mostly addressed to local optimization. A methodology of particular interest in this sense (Scheithauer et al. 2005; Stoyan and Chugay 2009; Stoyan et al. 1996, 2012) could well serve the scope of improving the approximate (quasi-global) solutions obtained with the approach discussed in this section. Stoyan's method introduces the concept of $\varPhi$-functions (e.g. Chernov et al. 2012; Stoyan et al. 2002, 2004). These are briefly outlined here below (limiting the discussion, for simplicity, to the two-dimensional case without rotations).

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

$\varPhi_{ij} > 0$ if $A_i(\boldsymbol{o}_i) \cap A_j(\boldsymbol{o}_j) = \{\emptyset\}$
$\varPhi_{ij} = 0$ if $\operatorname{int} A_i(\boldsymbol{o}_i) \cap \operatorname{int} A_j(\boldsymbol{o}_j) = \{\emptyset\}$ and $\partial A_i(\boldsymbol{o}_i) \cap \partial A_j(\boldsymbol{o}_j) \neq \{\emptyset\}$
$\varPhi_{ij} < 0$ if $\operatorname{int} A_i(\boldsymbol{o}_i) \cap \operatorname{int} A_j(\boldsymbol{o}_j) \neq \{\emptyset\}$

In such a way, $\varPhi_{ij} \geq 0$ guarantees that items $A_i(\boldsymbol{o}_i)$ and $A_j(\boldsymbol{o}_j)$ do not intersect (apart from their borders).

This section discusses the (two-dimensional) placement of simple polygons, i.e. polygons with no intersection between two nonconsecutive edges, inside a convex polygon. The approach introduced is closer, with respect to Stoyan's one, to alternative GO-based methodologies (e.g. Fischetti and Luzzi 2009; Sykora et al. 2011). An MINLP model is formulated. It is intended to be processed recursively, following a successive approximation philosophy. Once an acceptable (approximate) solution has been obtained, it can be exploited, as a starting point to solve the corresponding exact $\varPhi$-function-based MINLP model. The reader is referred to the previous work (Fasano 2013), for more details.

6.2.1 Approximate MINLP Model

Some necessary conditions, formulated in terms of MINLP constraints, are considered hereinafter. They are aimed at looking into approximate solutions to the two-dimensional problem of placing simple polygons (in the following just called polygons) from a given set I_P, into a convex polygon D (domain). The overall surface of the loaded items is maximized. For each polygon, any possible orientation is admitted. A recursive process is performed to improve, by successive approximation, the current solution, until a satisfactory one is reached. The positioning rules for each picked item are simply:

- *Each polygon has to be contained within D* (*domain* conditions).
- *Polygons cannot overlap* (*non-intersection* conditions).

To formulate the corresponding mathematical model, we shall consider a given (main) reference frame with origin O and axes w_β, $\beta \in \{1, 2\}$ (the set of axes is still denoted by B, cf. Sect. 2.1). The domain D is delimited by the set of vertices V_P, whose coordinates, with respect to the main reference frame, are represented by $V_{P\beta\gamma}$, $\gamma \in V_P$. They are assumed as nonnegative (without loss of generality). We shall then consider any polygon i (denoted in the following by P_i), from the given set I_P, and associate to it a local reference frame with origin $\boldsymbol{o}_{Pi}$, of coordinates $o_{P\beta i}$ (with respect to the main reference frame). The set of all vertices associated to polygon i is denoted by E_{Pi}. The coordinates of each vertex $\eta \in E_{Pi}$ are indicated, with respect to the local reference frame, by $V_{P\beta\eta i}$. The vector equations below hold

$$\forall i \in I_P, \forall \eta \in E_{Pi} \quad \boldsymbol{w}_{P\eta i} = \chi_{Pi}\boldsymbol{o}_{Pi} + \chi_{Pi}\left\|q_{\beta\beta'}\right\|_i \boldsymbol{V}_{P\eta i}. \tag{6.16}$$

Here for each vertex $\eta \in E_{Pi}$, $\boldsymbol{w}_{P\eta i} = (w_{P1\eta i}, w_{P2\eta i})^T$ is the vector of its coordinates with respect to the main reference frame; $\boldsymbol{o}_{Pi} = (o_{P1i}, o_{P2i})^T$; $\boldsymbol{V}_{P\eta i} = (V_{P1\eta i}, V_{P2\eta i})^T$; $\left\|q_{\beta\beta'}\right\|_i$ is the (orthogonal) rotation matrix of the local reference frame, with respect to the main one, and $\chi_{Pi} \in \{0, 1\}$, as in the previous cases, has the meaning: $\chi_{Pi} = 1$ if polygon i is picked; $\chi_{Pi} = 0$ otherwise.

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

$$\begin{aligned} &\forall \beta \in B, \forall i \in I_P, \forall \eta \in E_{Pi} \\ &w_{P\beta\eta i} = \sum_{\gamma \in V_P} V_{P\beta\gamma}\lambda_{P\gamma\eta i}, \end{aligned} \tag{6.17}$$

$$\forall i \in I_P, \forall \eta \in E_{Pi} \quad \sum_{\gamma \in V_P} \lambda_{P\gamma\eta i} = \chi_{Pi}, \tag{6.18}$$

where the variables λ are nonnegative and have the same meaning as in Sect. 2.1.

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

Proposition 6.1 *Given a set of internal circles C_{Pi} and C_{Pj}, for any pair of polygons i and j, respectively, no circle of C_{Pi} can intersect a circle of C_{Pj} .*

Proposition 6.2 *For any pair of polygons i and j, no vertex of P_i can belong to any circle of C_{Pj} and vice versa.*

Proposition 6.3 *For each pair of polygons i and j, any set of points of P_i must belong to the external region of P_j and vice versa: this holds in particular for all vertices of the polygons.*

Remark 6.4 In the above propositions, tangency conditions are admitted. In particular, it is understood that the external regions enclose the respective boundaries.

The third necessary *non-intersection* conditions posed above (Prop. 6.3) can be advantageously restricted to bounded external *slices*. To this purpose, the concept of *augmented polygon* is introduced by the following definition.

Definition 6.1 (Augmented polygon) For each P_{i}, consider the polygon, denoted by $\overline{P}_i$, such that

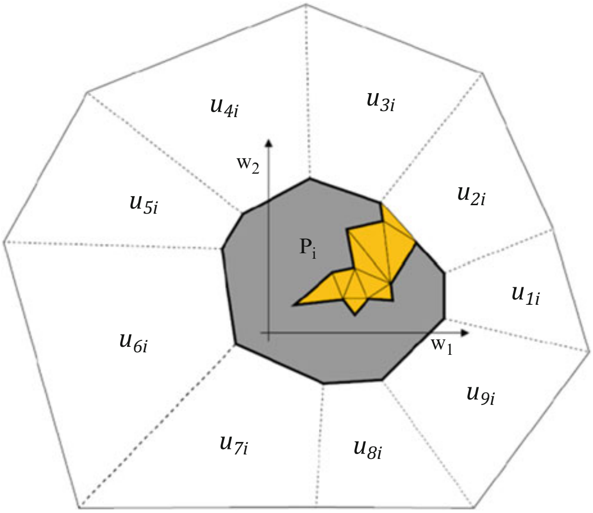


Fig. 6.3 Example of *augmented polygon*

$$P_i \subset \overline{P}_i,$$

$$\overline{P}_i - P_i = \bigcup_{\nu \in \overline{S}_i} U_{\nu i},$$

where $U_{\nu i}$ (slices) are convex polygon*s* (not necessarily disjoint), associated to P_i, and $\overline{S}_i$ is their set. Each $\overline{P}_i$ is called augmented polygon associated to P_i ($\overline{P}_i - P_i$ could always be partitioned into a set of triangles*;* see Fig. 6.3 and*,* for instance*,* de Berg et al. (2000) for polygon triangulation).

Figure 6.3 shows, as a matter of example, how a simple polygon can be augmented by convex slices. It is moreover immediately seen that whenever an internal 'cleft' (consisting of a non-convex simple polygon) is present, it can be partitioned into a set of triangles (i.e. convex *slices*). A specific case of the augmented polygon concept, adopted in the following, is provided by the definition below.

Definition 6.2 (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. It is denoted by $\overline{\overline{P}}_i$.

The third necessary *non-intersection* conditions (Prop. 6.3), when restricted to bounded external regions, can therefore simply be expressed as follows:

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* $\overline{\overline{P}}_j$ *and, vice versa, each point of* P_j *must belong to* $\overline{\overline{P}}_i$.

It is then immediately seen that Proposition 6.1 is expressed by the following constraints:

$$\begin{gathered}\forall i, j \in I_P / i < j, \forall h \in C_{Pi}, \forall k \in C_{Pj} \\ \sum_{\beta \in B} \left(o_{P\beta hi} - o_{P\beta kj}\right)^2 \geq \chi_{Pij} \left(R_{hi} + R_{kj}\right)^2.\end{gathered} \tag{6.19}$$

Here C_{Pi} and C_{Pj} denote the (arbitrary) sets of internal circles associated to polygons i and R_{hi} and R_{kj} the radius of the relative circles; $o_{P\beta hi}$ and $o_{P\beta kj}$ are their centre coordinates, with respect to the main reference frame; and the (implicitly binary) variables $\chi_{Pij} \in [0, 1]$ are subject to the same constraints expressed by (2.12a), (2.12b) and (2.13). The following vector equations hold:

$$\begin{gathered}\forall i \in I_P, \forall h \in C_{Pi} \\ \boldsymbol{o}_{Phi} = \chi_{Pi} \boldsymbol{o}_{Pi} + \chi_{Pi} \left\| q_{\beta\beta'} \right\|_i \boldsymbol{O}_{Phi}.\end{gathered} \tag{6.20}$$

They 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. Proposition 6.2 is very similar and it is not reported.

Given a *domain-covering augmented polygon* $\overline{\overline{P}}_i$, associated to polygon i, the corresponding set of *slices* are denoted by $\overline{\overline{S}}_i$. The set of vertices delimiting each *slice* ν (of $\overline{\overline{P}}_i$) is instead represented by $\overline{\overline{E}}_{\nu i}, \nu \in \overline{\overline{S}}_{i.}$ The following constraints express, for the polygon vertices, the third necessary *non-intersection* conditions (Prop. 6.3):

$$\begin{gathered}\forall \beta \in B, \forall i, j \in I_P, \forall \eta \in E_{Pi} \\ \chi_{Pij} w_{P\beta\eta i} = \sum_{\substack{\gamma \in \overline{\overline{E}}_{\nu j}, \\ \nu \in \overline{\overline{S}}_j}} \lambda_{P\eta i \gamma \nu j} \ w_{P\beta\gamma\nu j},\end{gathered} \tag{6.21}$$

$$\begin{gathered}\forall i, j \in I_P, \forall \eta \in E_{Pi}, \forall \nu \in \overline{\overline{S}}_j \\ \sum_{\gamma \in \overline{\overline{E}}_{\nu j}} \lambda_{P\eta i \gamma \nu j} = \chi_{P\eta i \nu j},\end{gathered} \tag{6.22}$$

$$\forall i, j \in I_P, \forall \eta \in E_{Pi} \quad \sum_{\nu \in \overline{\overline{S}}_j} \chi_{P\eta i \nu j} = \chi_{Pij}, \tag{6.23}$$

where, as before, $w_{P\beta\eta i}$ are the coordinates of polygon i vertices with respect to the main reference frame. Similarly, $w_{P\beta\gamma\nu j}$ are the vertex coordinates of *slices* ν associated to polygon j; $\lambda_{P\eta i \gamma \nu j}$ are nonnegative variables and $\chi_{P\eta i \nu j} \in \{0, 1\}$. Constraints (6.21), (6.22) and (6.23) ensure thus that if both polygons i and j are loaded, then each vertex of polygon i will belong to one *slice* ν of (the *augmented polygon* associated to) j and vice versa.

Remark 6.5 As is easily gathered, the presence of the $\chi_{P\eta i \nu j}$ binary variables increases the model complexity dramatically. Constraints (6.22) and (6.23) could thus be profitably substituted by the following $\forall i, j \in I_P, \forall \eta \in E_{Pi}, \forall \nu \in \overline{\overline{S}}_j \quad \sum_{\gamma \in \overline{\overline{E}}_{\nu j}} \lambda_{P\eta i \gamma \nu j} = \chi_{Pij}$.

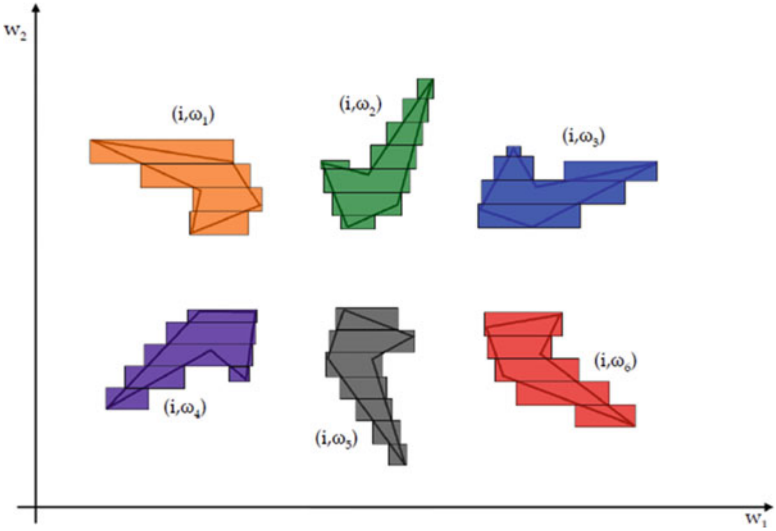


Fig. 6.4 Covering tetris-like items (corresponding to possible rotations $\omega_1 - \omega_6$, of the same polygon i)

The logical restriction expressed by constraints (6.23) may indeed be treated algorithmically, by introducing appropriate special ordered sets, similar to those suggested by Escudero (1988). More specifically, in such a case, only the variables $\lambda_{P\eta i\gamma\nu j}$ corresponding to a single slice $\nu \in \overline{\overline{S}}_j$ would be allowed to be positive, while all remaining are forced to zero.

6.2.2 *Applications*

The difficulty in solving the MINLP model of Sect. 6.2.1 is, *per se*, extremely high, even when small-scale instances are involved. Consequently, it is not expected to accomplish the task directly, and an incremental procedure is strongly recommended. As a rough approximation, for instance, for each polygon i, just one of its biggest internal circles could be considered. The number of internal circles could be sequentially increased, for all pairs of polygons currently intersecting, until a satisfactory (approximate) solution is attained. And similar considerations hold for all the necessary conditions considered in Sect. 6.2.1.

As, in any case, the MINLP solution process efficiency is strongly affected by the initial guess available, the tetris-like model of Sect. 2.1 can be utilized for this purpose. This can be done by temporarily replacing the given polygons with covering tetris-like items and considering, for each polygon $i_P \in I_P$, just a set Ω_{Pi} of possible (arbitrary) discretized rotations; see Fig. 6.4.

For each polygon $i_P \in I_P$ and each selected rotation, $\omega_{Pi} \in \Omega_{Pi}$, let us define a single tetris-like item, covering the polygon for that rotation and such that its sides are orthogonal/parallel to the main reference frame axes. Denoting by I_T the set of all tetris-like items built in such a way, the one-to-one correspondence $(i_P, \omega_{Pi}) \in I_P \times \Omega_{Pi} \leftrightarrow i_T \in I_T$ is defined. This way, the subset of tetris-like items I_{Ti_P} is associated to each polygon $i_P \in I_P$. The problem in question is hence that of placing the covering tetris-like items of set I_T, one (and only one) for each subset I_{Ti_P}, into D, without any possibility of rotation. This leads to a special case of the MIP model of Sect. 2.1: constraints (2.1) are dropped, whilst (2.2), (2.3), (2.4), (2.5a), (2.5b) and (2.6) are kept, setting all variables θ to one and eliminating all indexes ω, as well as the related sums. Let us denote (with a little of abuse of notation) by $\chi_{i_P} \in \{0,1\}$ and $\chi_{i_T} \in \{0,1\}$ the decisional variables controlling the selection of polygon $i_P \in I_P$ and its associated (pre-oriented) tetris-like item $i_T \in I_{Ti_P}$. The following conditions have to be added:

$$\forall i_P \in I_P \quad \sum_{i_T \in I_{Ti_P}} \chi_{i_T} = \chi_{i_P}. \tag{6.24}$$

This guarantees that if polygon i_P is picked, it is represented by one and only one tetris-like item i_T, corresponding to a specific orientation (from the set of the discretized ones associated to i_P).

Remark 6.6 An alternative tetris-like item approximation could be considered, keeping all the constraints (2.1), (2.2), (2.3), (2.4), (2.5a), (2.5b) and (2.6) (formally) as they are. Each projection $L_{\omega\beta hi}$ would be associated to a specific (discretized) orientation of the related polygon. These terms, however, would no longer correspond to rigid rotations of the relative tetris-like items (covering tetris-like items, indeed, change their pattern, depending on the orientation). Figure 6.4 clearly illustrates this aspect.

With the approximation approach suggested, 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 thus represents (for the surface maximization) a lower bound. Once a good initial solution is attained, the actual items (polygons) can be introduced and the MINLP process activated.

A heuristic dedicated to the polygon packing problem is currently at a prototyping stage (see Fasano 2013; LGO Solver Suite for Global–local Nonlinear Optimization is utilized as a nonlinear optimizer, see Pintér 1997, 2002, 2005, 2007; Pintér Consulting Services 2013). It performs the tetris-like approximation as an initialization step (*virtual* items are purposely introduced, in order to concentrate unexploited areas in a limited number of uncovered zones). Then, different general techniques, such as item fixing/exchange and ‘hole’ filling, are adopted (exploiting the features of Prop. 6.1, 6.2 and 6.34). An experimental investigation is currently under study (Fasano 2013).

Concerning the computational aspects related to the first phase, based on a two-dimensional tetris-like item MIP model, insights can be derived from Chap. 5.

Fig. 6.5 Domain with internal polygonal 'holes'

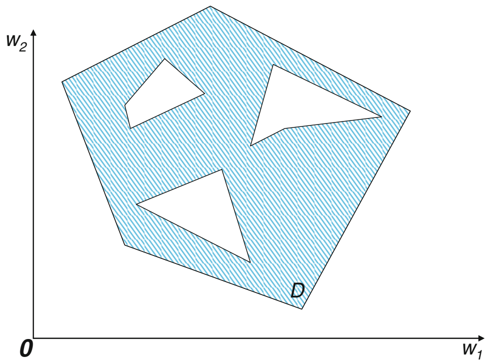
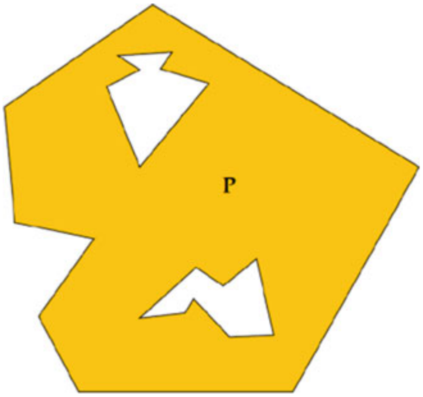


Fig. 6.6 Simple polygons with 'holes'



The necessary conditions expressed by Propositions (6.1) and (6.2) rely, instead, on the MINLP formulation of *circle packing*, and significant topical literature is available (e.g. Castillo et al. 2008; Hifi and M'Hallah 2009; Specht 2012).

It is argued that the additional conditions discussed in Sect. 2.1 could also be (at least in principle) extended to the case of polygon packing. This holds, for instance, when balancing restrictions are stated. Domains with 'holes' or forbidden zones can in general be modelled by introducing zero-mass items (see Fig. 6.5). Moreover, when some items contain 'holes', these become part of their external regions (see Fig. 6.6).