

Chapter 4

Heuristic Approaches for Solving the Tetris-like Item Problem in Practice

As easily gathered, the general MIP model, conceived to sort out the *tetris*-like item packing problem (Sect. 2.1), is usually very hard to solve. In this chapter the relevant intrinsic difficulties are examined first (Sect. 4.1). A heuristic philosophy is then emphasized to tackle efficiently the problem, even if just nonproven optimal solutions can, in general, be obtained.

The basic concept of *abstract configuration* is introduced (Sect. 4.2). Chapter 3 reformulations, devised to solve the *feasibility* subproblem, are exploited to look into an initial approximate solution (Sect. 4.3.1). Two alternative heuristic procedures, thought up to improve it recursively, until a satisfactory result is reached, are discussed next (Sects. 4.3.2 and 4.3.3). The possibility of interacting with the solution process is also outlined (Sect. 4.3.4).

4.1 Intrinsic Difficulties

Broad classes of packing problems are notoriously well known for being *NP-hard* (as regards this classification and, more in general, the theoretical aspects related to *computational complexity* and *approximability*, see, for instance, Ausiello et al. 2003, Chlebík and Chlebíková 2006, Goldreich 2008).

The general MIP model reported in Chap. 2 is, per se, extremely hard to solve in practice, when real-world instances have to be dealt with. This holds, even if only single parallelepipeds are involved and no additional conditions are set (see Sect. 2.1, special case). In this circumstance, insights on its complexity can be provided by looking upon the model overall structure (N indicates here the total number of single parallelepipeds):

$O(3N(N - 1))$ binary variables σ

$O(9N)$ binary variables δ

$O(N)$ binary variables χ

$O(6N)$ *orthogonality* constraints

$O(3N)$ domain constraints

$O\left(7\frac{N(N-1)}{2}\right)$ non-intersection constraints (of which $O(3N(N-1))$ big- M s)

The model scale increases quite dramatically when N actual tetris-like items are involved. In order to discuss this case, we shall indicate, for each item i , with $|C_i|$ the cardinality of the set C_i relative to its *components*. The total number of pairs of *components*, belonging to different items, is denoted by N_C and computed as follows:

$$N_C = \left(\sum_{i \in I} |C_i| \right) - \sum_{i \in I} \binom{|C_i|}{2}, \quad (4.1)$$

where the symbol $\binom{N_1}{N_2}$ represents, in general, the N_2 -combinations of a set containing N_1 -elements. The model overall structure is hence represented by the following:

$O(24N)$ binary variables θ

$O(N)$ binary variables χ

$O(6N_C)$ binary variables σ

$O(7N_C)$ non-intersection constraints (of which $O(6N_C)$ big- M s).

Even at a first glance, the intricacy of both scenarios mentioned above is not only related to the binary variables. Indeed, it primarily depends on the presence of the *big- M* constraints, related to the *non-intersection* conditions. The occurrence of the implicit *transitivity* conditions (see Sect. 3.2), moreover, provides a significant insight on the hidden model difficulties. The following proposition shows how their total number can be computed.

Proposition 4.1 *Given N tetris-like items, the total number of transitivity conditions is* $18 \sum_{\substack{i, i', i'' \in I \\ i < i' < i''}} |C_i| \cdot |C_{i'}| \cdot |C_{i''}|$.

Proof To prove this proposition we can firstly concentrate on the particular case of Sect. 3.2, where the sequence h precedes h' and h' precedes h'' ($h < h' < h''$) along the (general) axis w_β is considered. The conditions $\forall \beta \in B, \forall i, i', i'' \in I / i < i' < i'', \forall h \in C_i, \forall h' \in C_{i'}, \forall h'' \in C_{i''} \sigma_{\beta h h' i''}^- \geq \sigma_{\beta h h' i''}^- + \sigma_{\beta h' h'' i''}^- - 1$ are therefore recalled.

Selecting (for the time being) the specific axis w_β , for a given pair of components h and h' of i and i' , respectively, there are $|C_{i''}|$ such conditions; for a given component h of i , they are $|C_{i'}| \cdot |C_{i''}|$, so that their total number is $|C_i| \cdot |C_{i'}| \cdot |C_{i''}|$. This corresponds to the order relation $(h \in C_i) < (h' \in C_{i'}) < (h'' \in C_{i''})$. There are 3! permutations of such a kind, for which the above transitivity conditions have to be

properly rearranged (utilizing appropriately the variables σ^- and σ^+). This must hold for any triplet i, i' and i'' , such that $i < i' < i''$, and for each axis w_β . \square

Remark 4.1 When all the tetris-like items involved have the same number ($|C_i|$, $\forall i \in I$) of components, the total number of transitivity conditions is $18|C_i|^3 \binom{N}{3}$.

To consider a quantitative example, an instance involving 50 items, of five *components* each, the order of magnitude relevant to all the *non-intersection (big-M)* constraints is $6 \times 3 \times 10^4$; that of the *transitivity* conditions is 4×10^7 . Section 4.2 investigates the sets of variables σ^+ and σ^- (at most one for each pair of components belonging to different items) that, if fixed to one, are compliant with the *transitivity* implications. These variables σ are called *transitivity compatible*.

4.2 Abstract Configurations

The *abstract configuration* concept is introduced by the following definitions.

Definition 4.1 Constraints of the types

$$w_{\beta 0hi} - w_{\beta 0kj} \geq \frac{1}{2} \sum_{\omega \in \Omega} (L_{\omega\beta hi} \vartheta_{\omega i} + L_{\omega\beta kj} \vartheta_{\omega j}),$$

$$w_{\beta 0kj} - w_{\beta 0hi} \geq \frac{1}{2} \sum_{\omega \in \Omega} (L_{\omega\beta hi} \vartheta_{\omega i} + L_{\omega\beta kj} \vartheta_{\omega j}),$$

corresponding to either $\sigma_{\beta hkij}^+ = 1$ or $\sigma_{\beta hkij}^- = 1$ in (2.5a) and (2.5b), respectively, are called relative position constraints.

Definition 4.2 Given a set of N items and all the N_C pairs of components, belonging to different items, an abstract configuration consists of N_C relative position constraints (one and only one for each pair) compatible in any unbounded domain.

Remark 4.2 From Definition 4.2, it follows immediately that, for a given set of N items, an abstract configuration corresponds to a set of N_C transitivity-compatible σ variables fixed to one.

To interpret the concept of *abstract configuration*, it is opportune to restrict the discussion to the simpler case of single-*component* items (see Sect. 2.1, special case). Theoretically, in such a case, all the *abstract configurations*, associated to a given set of N items, could be directly derived by adopting a three-dimensional generalization of the *square grid graph* (e.g. Weisstein 2012). This is the graph whose vertices correspond to the points of $\mathcal{N}^3 \subset \mathcal{R}^3$ (referred to an orthogonal reference frame), with integer coordinates that are in the range $0, \dots, N-1$ and with any two vertices connected by an edge, whenever they are at unit distance; see Fig. 4.1.

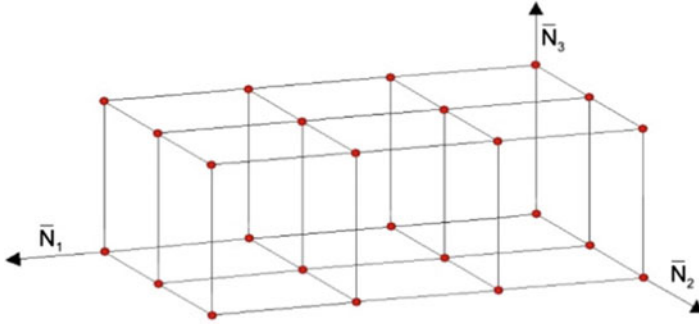


Fig. 4.1 Three-dimensional grid

Since the *relative positions* are to be considered in an unbounded domain, from this ‘topological’ perspective, the actual dimensions of the items can be totally neglected. Items can, thus, simply be considered as geometrical points and their relative placement represented by N nodes (intended as points of N^3) of the aforementioned grid. A set of corresponding *relative position* constraints can hence be selected to generate an *abstract configuration*. If, for instance, item i is associated to node $(0,0,0)$ and item j is associated to node $(N - 1, N - 1, N - 1)$, their coordinate relative distance on the grid, with respect to each axis, is $(N - 1)$ units and i precedes j in all directions. This, in an unbounded domain, is compliant, with the following *relative position* constraints:

$$w_{1j} - w_{1i} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 1i} + L_{\alpha j} \delta_{\alpha 1j}),$$

$$w_{2j} - w_{2i} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 2i} + L_{\alpha j} \delta_{\alpha 2j}),$$

$$w_{3j} - w_{3i} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 3i} + L_{\alpha j} \delta_{\alpha 3j}).$$

Any of the above *relative position* constraints can thus be selected, as the one corresponding to the pair of items (i, j) , to create an *abstract configuration* (relative to the given set of N items). Similar considerations hold for all the possible $\binom{N^3}{N}$ associations of N items to the N^3 grid nodes. It is worth noticing that, in any unbounded domain, each given *abstract configuration* yields an infinity of packing scenarios, obtainable by simple roto-translations of the items. This suggests that even when an *abstract configuration* is forced, if compliant with the given domain, the items still have a certain freedom of movement.

As it is immediately understood, even for relatively small-scale instances, the number of all possible associations of N items to N^3 grid nodes is immense.

Of course, it is higher than that of the *abstract configurations* which can actually be generated, as the following example shows. Let us consider the set of three items i , i' and i'' , together with the two following distinct associations A_1 and A_2 :

$$\begin{aligned} A_1) \quad & i \rightarrow (0, 0, 0), \quad i' \rightarrow (2, 0, 0), \quad i'' \rightarrow (1, 2, 0); \\ A_2) \quad & i \rightarrow (0, 1, 0), \quad i' \rightarrow (2, 1, 0), \quad i'' \rightarrow (1, 2, 0). \end{aligned}$$

These are compliant with the same *relative position* constraints and, thus, give rise to the same corresponding *abstract configurations*.

Considering also that some associations ‘dominate’ others, in the sense that their corresponding sets of *abstract configurations* strictly contain those of the ‘dominated’ ones, several further redundancies are expected. It is gathered that quite a lot of duplications could be eliminated. The number of grid nodes could be reduced, for instance, taking into account the actual size of the domain D . Indeed (see Fig. 4.1), its maximum value, on each axis w_β , respectively, is given by

$$\forall \beta \in B \quad \bar{N}_\beta = \max_{\substack{N' \leq N \\ \sum_{i=1}^{N'} L_{i\beta} \leq D_\beta}} \{N'\}.$$

It is, nevertheless, obvious that any exhaustive approach, based on the generation of all possible *abstract configurations*, would be impracticable, in real-world cases. It is, moreover, quite intuitive that a generalized use of the three-dimensional grid, to cope also with the case of tetris-like items (and not only of single parallel-epipeds), would be quite tricky indeed.

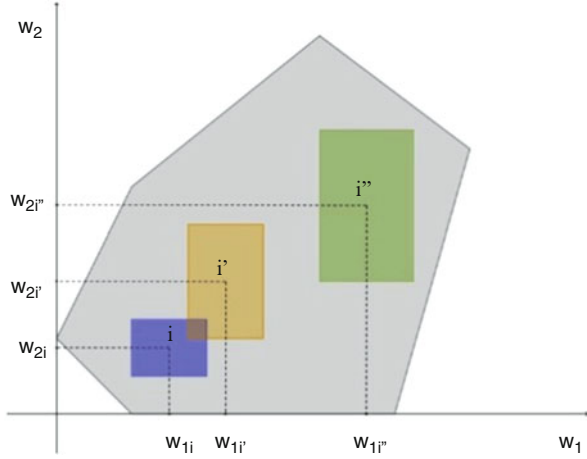
The basic idea of the heuristic approaches put forward in Sect. 4.3 addresses instead the exploitation of a number of ‘good’ *abstract configurations*. The following discussion focuses therefore on the capability of extracting, from approximate solutions (with possible item overlap), *abstract configurations* that, at least partially, consider the actual characteristics of the problem (e.g. conditions on items, domain and balancing). The thus obtained *abstract configurations* are then exploited to give rise to *integer-feasible* (even if, generally, suboptimal) solutions, compliant with all the given conditions (including, most certainly, *orthogonality*, *domain* and *non-intersection*).

Given any approximate solution, the objective of the *abstract configuration* generation consists of assigning one and only one *relative position* constraint to each pair of *components*, belonging to different items (in the following, this is understood, when referring to any pair of *components*).

The (arbitrary) rules listed below represent very simple selection criteria the *abstract configuration* generation could be based on; see Fig. 4.2:

- If the *components* do not intersect and the *non-intersection* conditions hold only with respect to one axis, the corresponding *relative position* constraint is selected.

Fig. 4.2 *Abstract configuration* derived from an approximate solution (two-dimensional representation of single-component items)



- If the *components* do not intersect and the *non-intersection* conditions hold for more than one axis, the *relative position* corresponding to the maximum distance between the projections of the respective coordinate centres is selected.
- If the *components* intersect, the *relative position* corresponding to the maximum distance between the projections of the respective coordinate centres is selected.

Figure 4.2 provides a (two-dimensional) representation of the above selection rules. Just single-*component* items, denoted as i , i' and i'' , respectively, are considered, to make the example clearer (the generalization to actual tetris-like items is straightforward). As easily seen in the figure, they are subject to the following conditions:

- Items i and i' projections overlap on both the axes w_1 and w_2 , with $w_{2i'} - w_{2i} > w_{1i'} - w_{1i}$.
- Items i and i'' projections neither overlap on the axis w_1 nor on w_2 , with $w_{1i''} - w_{1i} > w_{2i''} - w_{2i}$.
- Items i' and i'' projections do not overlap on the axis w_1 only, with $w_{1i''} > w_{1i'}$.

Based on the rules listed above, the following *abstract configuration* (relative to the single-*component* items i , i' and i'') is extracted:

$$w_{2i'} - w_{2i} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 2i} + L_{\alpha j} \delta_{\alpha 2i'}),$$

$$w_{1i''} - w_{1i} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 1i} + L_{\alpha j} \delta_{\alpha 1i''}),$$

$$w_{1i''} - w_{1i'} \geq \frac{1}{2} \sum_{\alpha \in A} (L_{\alpha i} \delta_{\alpha 1i} + L_{\alpha j} \delta_{\alpha 1i''}).$$

Remark 4.3 As the above selection criteria are not based on rigorous reasoning, it is evident that a number of different (and supposedly more sophisticated) empirical rules could be explored. In case of intersection, the third rule proposed above could be, for instance, substituted with the one that selects the relative position corresponding to the projection where the minimum overlap occurs.

4.3 Solution Search

As already pointed out, the generation of *abstract configurations*, on the basis of approximate solutions, represents a fundamental concept of the overall philosophy adopted here. The first step of both the heuristic processes that are to be addressed in this section focuses, therefore, on the creation of a high-quality starting approximate solution. It is trusted, indeed, that the ‘closer’ to an actual solution it is, the ‘better’ the generated *abstract configuration* results. Consequently, less computational effort is needed to obtain a satisfactory ultimate result. This step is denoted as *initialization*.

The heuristic processes outlined in Sects. 4.3.2 and 4.3.3, respectively, provide, as a matter of fact, two alternative search strategies to work out (at least at a suboptimal level) the general MIP model (with possible additional conditions). As an overall rule of thumb, the first is more oriented to solving quite tricky instances but with a relatively limited total number (<100) of item *components* involved. The second is mainly proposed for larger ones. A joint use of the two (also in a parallelized mode) could be subject to further investigation.

4.3.1 Initialization

The reformulated models of Sect. 3.1, aimed at solving the *feasibility* subproblem, can properly be adapted to serve the *initialization* purpose. Partial *LP relaxations*, either of the first (Sect. 3.1.1) or the second (Sect. 3.1.2) linear reformulations (including all the given additional conditions or a subset of them), are hence utilized. They have the scope of finding a first approximate solution, to be refined (if opportune) by the nonlinear reformulation (Sect. 3.1.4).

In all such approximate solutions, the overlap of items is allowed, but it is minimized, in the sense specified, for each reformulated version, in Sect. 3.1. On the contrary, both the *orthogonality* and *domain* constraints are imposed. Of course, the more the additional conditions are included (even if in approximate versions), the more the obtained starting solution is a realistic representation of the actual problem to solve.

Hereinafter, we shall reexamine, in this perspective, the three reformulations, one at a time. In all these cases, we shall still consider the problem in terms of *feasibility*, as it will be tentatively assumed, a priori, that all the given items can be loaded.

This way, all variables χ are set to one and the constraints involving them readjusted properly. In each reformulation, as stated above, constraints (2.1), (2.2), (2.3) and (2.4) are maintained. In the following discussion, for the sake of simplicity, the additional conditions are neglected, as it is understood that they can be properly added each time, accordingly to the specific problem to deal with.

4.3.1.1 Use of the First Linear Reformulation

A partial *LP relaxation* of the first linear reformulation is carried out by eliminating inequalities (3.2a), (3.2b) and (3.3), whilst keeping (3.1a), (3.1b) and the *objective function* (3.4) unaltered.

Remark 4.4 It should be observed that, even when no additional conditions are present, the (partially) *LP-relaxed* model is still of an MIP type. This is due to the presence of the orientation variables. When the number of components involved is very high, the orientation of some (or all) items could be pre-fixed (on the basis of some empirical criterion). Analogously, it could be done for the biggest side.

4.3.1.2 Use of the Second Linear Reformulation

The second linear reformulation can be adopted, as an alternative to the first. The *non-intersection* constraints (3.5a) and (3.5b) are kept the same, whilst, as partial *LP relaxation*, inequalities (3.6) are dropped. The *objective function* (3.7) is maintained as is. This way, whilst the model is significantly simplified, it is no longer guaranteed that each component is actually enclosed by the associated parallelepiped (whose volume is maximized, in the sense specified in Remark 3.2).

If, moreover, all *components*, once reduced to the corresponding cubes of sides L_{1hi} , can actually be loaded into D , the following bounds are advantageously imposed. They are aimed at mitigating the mutual competition among items, i.e. avoiding that excessive volume is attributed to some *components* to the detriment of others:

$$\forall \beta \in B, \forall i \in I, \forall h \in C_i \quad L_{1hi} \leq l_{\beta hi} \leq L_{3hi}. \quad (4.2)$$

Otherwise, if not all the cubes of sides L_{1hi} can be loaded, the lower bounds appearing in (4.2) may be properly reduced, for instance, by subsequent attempts. Alternatively, they could be rewritten as $L_{1hi} - r_{\beta hi} \leq l_{\beta hi}$, where the terms $r_{\beta hi}$ are non-negative variables. In such a case, the *objective function* (3.7) can be substituted with $\max_{\substack{\beta \in B, \\ i \in I, h \in C_i}} (l_{\beta hi} - K_R r_{\beta hi})$, where K_R represents an appropriate

(positive) coefficient.

It should, moreover, be noticed that the approximate solution obtained by the second linear reformulation, when the (lower) bounds (4.2) are introduced, provides a set of *transitivity-compatible* variables σ (relative to all the items involved). Since they already represent an *abstract configuration*, no further generation of it is needed.

If the model variation outlined at the end of Sect. 3.1.2 (substituting (3.6) with (3.8)) is to be applied, a possible ‘relaxation’ can simply be obtained by renouncing the global optimal solution (cf. Sect. 4.3.1.3) and thus admitting possible intersections among items (quite satisfactory suboptimal solutions are expected to be found with a moderate computational effort).

4.3.1.3 Use of the Nonlinear Reformulation

As already mentioned, this model is aimed at improving the approximate solutions obtained either by the first or second reformulations. The MIP solution obtained by either the first or second linear reformulation is then inherited to initialize its related Mixed Integer Non-Linear Programming (MINLP) search process.

In this case, the ‘relaxation’ just concerns the solution quality, in the sense that no global optimal solution is requested. Looking for local optimal solutions only, indeed, means trying to minimize the item overall overlap, but, obviously, it is not guaranteed that the *non-intersection* constraints are satisfied.

Remark 4.5 A possible alternative to the above nonlinear reformulation consists of substituting, in the first linear one, the objective function (3.4), with the following:

$$\max \sum_{\substack{\beta \in B, \\ i, j \in I / i < j, \\ h \in C_i, k \in C_j}} \frac{\left(d_{\beta h k i j}^+ - d_{\beta h k i j}^-\right)^2}{D_{\beta}^2}.$$

This way, the item overall intersection is no longer minimized with direct ‘competition’ between each pair of variables $d_{\beta h k i j}^+$ and $d_{\beta h k i j}^-$, relative to the same reference frame axis w_{β} . Nonetheless, it should be noticed that, with this formulation, even the attainment of a global optimum is not up to guaranteeing that no intersection occurs.

Remark 4.6 It should be noticed that the total superposition of the centres of two components can occur both with the first linear reformulation and the second one, when the (lower) bounds (4.2) are neglected. In this case, the approximate solution can properly be perturbed, in order to get rid of this ambiguity (with the rules presented, for instance, in Sect. 4.2, to generate the abstract configuration, indeed, in case of such superposition, no relative position could be selected). This aspect is, however, not considered here, for the sake of simplicity.

Remark 4.7 It is gathered that, as a first (and quite daring) attempt, the abstract configuration derived from the initialization step could be directly forced into the general MIP model, requesting that all items are loaded. To do this, it would be sufficient to substitute all the non-intersection constraints (2.5a), (2.5b) and (2.6) with the available relative position ones. It is obvious, however, that an integer-feasible solution could hardly be found this way, even for very simple instances.

Remark 4.8 As some items may have prefixed orientation, this aspect could be taken into account also during the initialization step (within the limits implied by its characteristic of admitting overlap). When utilizing the first linear reformulation or the nonlinear one, the imposition of the given pre-orientations is straightforward. The situation is more complicated when the second linear reformulation is adopted. In such a case, the related model can be properly modified.

Let us suppose, for the sake of simplicity, that all items are pre-oriented. Inequalities (3.5a) and (3.5b) can be modified as follows:

$$\begin{aligned} \forall \beta \in B, \forall i, j \in I / i < j, \forall h \in C_i, \forall k \in C_j \\ w_{\beta 0hi} - w_{\beta 0kj} &\geq \frac{1}{2} \left(L'_{\beta hi} - l'_{\beta hi} + L'_{\beta kj} - l'_{\beta kj} \right) - D_{\beta} \left(1 - \sigma_{\beta hkij}^+ \right), \\ \forall \beta \in B, \forall i, j \in I / i < j, \forall h \in C_i, \forall k \in C_j \\ w_{\beta 0kj} - w_{\beta 0hi} &\geq \frac{1}{2} \left(L'_{\beta hi} - l'_{\beta hi} + L'_{\beta kj} - l'_{\beta kj} \right) - D_{\beta} \left(1 - \sigma_{\beta hkij}^- \right). \end{aligned}$$

Here, for each *component* h of item i , the terms $L'_{\beta hi}$ represent its pre-oriented sides, and $l'_{\beta hi}$ are non-negative variables. The *objective* function $\min \sum_{\substack{\beta \in B, \\ i \in I, h \in C_i}} l'_{\beta hi}$ can

then substitute (3.7) and bounds (4.2) have to be properly rewritten.

Remark 4.9 When the number of items/components involved is large, the initialization step, independently from the formulations adopted, may require quite a significant computational effort. To cope with this practical difficulty, the items involved in the original instance can be partitioned in subsets and added incrementally. At each step, the abstract configuration corresponding to the items already loaded is imposed (taking appropriate precautions to prevent possible infeasibilities). Once a new subset of items has been added, the corresponding solution gives rise to an upgraded abstract configuration, relevant to all the so-far-loaded items.

Notice that a similar recursive approach could advantageously be adopted by utilizing the non-restrictive reformulation of Sect. 3.1.3. Whenever it stops yielding improvements, either the first or the second linear reformulations can take its place in the process, in order to eventually include all the items of the original instance. It is understood that the use of the non-restrictive reformulation could be quite advantageous, since, for each (suboptimal) solution found, no overlap can occur.

4.3.2 Heuristic Process Based on Suggested Abstract Configurations

The heuristic process outlined here stresses the idea of inheriting the *abstract configuration* derived from the *initialization* step (Sects. 4.2 and 4.3.1). It is exploited (by subsequent adaptations) to obtain a satisfactory solution to the general MIP model (inclusive of the additional conditions if present, see Sect. 2.3). The *abstract configuration* and its modifications, achieved during the whole procedure, are ‘suggested’ recursively, adding or subtracting, time after time, items from the given set. This is carried out by means of a dichotomous approach.

In order to ‘suggest’ any *abstract configuration*, the general MIP model *objective* function (2.7) has to be properly modified. For this purpose, let us firstly recall that, as pointed out in Remark 4.2, any *abstract configuration* can be represented by a set of (*transitivity-compatible*) variables σ^+ and σ^- , fixed to one. This way, on the basis of the *abstract configuration* that is to be ‘suggested’, the following coefficients $\hat{\sigma}_{\beta hkij}^+$ and $\hat{\sigma}_{\beta hkij}^- \in \{0, 1\}$ are introduced:

$$\forall \beta \in B, \forall i, j \in I / i < j, \forall h \in C_i, \forall k \in C_j$$

$\hat{\sigma}_{\beta hkij}^+ = 1$ if the *relative position* constraint $w_{\beta 0hi} - w_{\beta 0kj} \geq \frac{1}{2} \sum_{\omega \in \Omega} (L_{\omega \beta hi} \vartheta_{\omega i} + L_{\omega \beta kj} \vartheta_{\omega j})$ belongs to the *abstract configuration*; $\hat{\sigma}_{\beta hkij}^+ = 0$ otherwise.

Analogous constraints can be stated for $\hat{\sigma}_{\beta hkij}^-$. The *objective* function below substitutes then (2.7):

$$\max \sum_{\substack{\beta \in B, \\ i, j \in I / i < j, \\ h \in C_i, k \in C_j}} \left(\hat{\sigma}_{\beta hkij}^+ \sigma_{\beta hkij}^+ + \hat{\sigma}_{\beta hkij}^- \sigma_{\beta hkij}^- \right). \quad (4.3)$$

As is easily realized, this new optimization criterion has the effect of inducing each variable σ , corresponding to a non-zero $\hat{\sigma}$, to attain the value of one, in compliance with the ‘suggested’ *abstract configuration*. This has the expected effect of obtaining an *integer-feasible* solution, when existing, by a dramatically reduced computational effort.

In order to perform the heuristic process under consideration here, a first attempt is made by imposing that all the given items must be loaded. The *abstract configuration* derived from the *initialization* step is thus adopted. If any *integer-feasible* solution to the general MIP model, modified with the ad hoc *objective* function (4.3), is obtained, the given problem is solved. In the opposite case, the following dichotomous procedure is started.

An initial subset of all the given items is taken, by imposing (still tentatively) that all of them must be loaded. If any *integer-feasible* solution is obtained, a new *abstract configuration* (relative to the whole set of items) is derived, simply by carrying out a new *initialization* step: for all the items already loaded, their *relative positions* are fixed (when the nonlinear reformulation is used, the model includes them, as explicit constraints). A new *abstract configuration* is then generated.

In the opposite case, i.e. when the above subset of items does not allow for any *integer-feasible* solution, some of them are rejected, until a successful result is obtained. The heuristic process is then executed by activating, recursively, either the forward or the backward steps just described. Since the process tries and modifies, time after time, the ‘suggested’ *abstract configurations* as little as possible, the whole effect is that of performing an overall *depth-first* strategy.

Remark 4.10 Different versions of the heuristic approach described in this section could of course be considered. When ‘suggesting’ the current abstract configuration, for instance, some of the corresponding relative position constraints could be tentatively forced.

4.3.3 Heuristic Process Based on Imposed Abstract Configurations

A heuristic procedure, based on the imposition of *abstract configurations*, has been previously introduced (Fasano 2008) to tackle instances involving only single parallelepipeds to load into a convex domain. The adopted approach has been refined and extended to the case of actual tetris-like items, as outlined hereinafter.

This process, summed up by Fig. 4.3, is aimed at generating and imposing a sequence of ‘good’ *abstract configurations* and at solving correspondingly, step by step, a reduced MIP model, until a final satisfactory solution is attained.

The reduced model (inclusive of the additional conditions, when present; see Sect. 2.3) is derived, each time, from the general one (see Sect. 2.1) by eliminating all the redundant *non-intersection* constraints and variables σ , not contemplated by the *abstract configuration* imposed.

The *packing* module (see Fig. 4.3) is assigned the task of solving, time after time, the reduced models. Items are rejected, if necessary, to make the current *abstract configuration* (referred to the whole set of the given items) compatible with the given domain D (and the additional conditions, if any). The *item-exchange* and *hole-filling* modules (see Fig. 4.3) are employed, during the whole process, to provide new approximate solutions and supposedly improved *abstract configurations*. The *abstract configurations* are generated (on the basis of what is discussed in Sect. 4.2) from the approximate solutions obtained either by the *initialization* (Sect. 4.3.1) or the *hole-filling* steps. When derived from this one, the *relative positions* of the items previously loaded by the *packing* module are contemplated. The *item-exchange* step directly provides a new *abstract configuration*, so that no

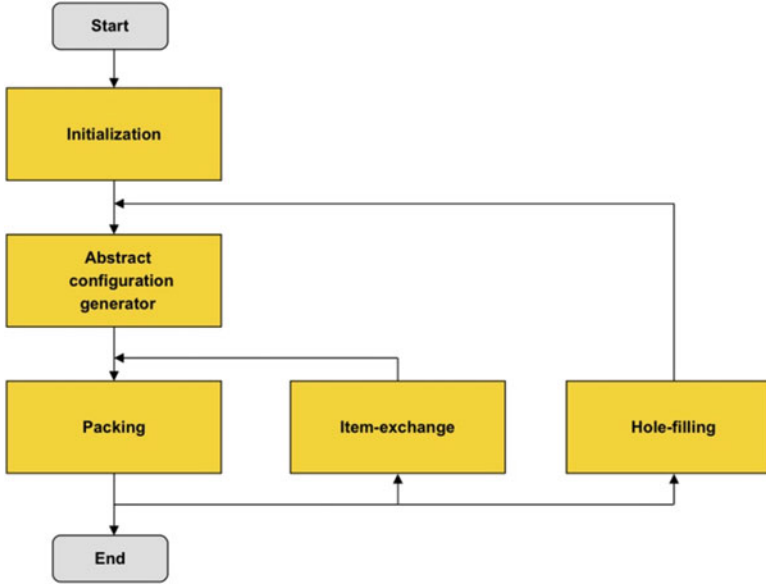


Fig. 4.3 Heuristic process based on the *abstract configuration* imposition

proper generation activity has to be performed. The *packing* module is discussed hereinafter, together with those dedicated to the *item-exchange* and *hole-filling* steps, neglecting several details, unnecessary for an overall comprehension. The general methodology is further outlined, suggesting a possible alternative approach.

4.3.3.1 Packing

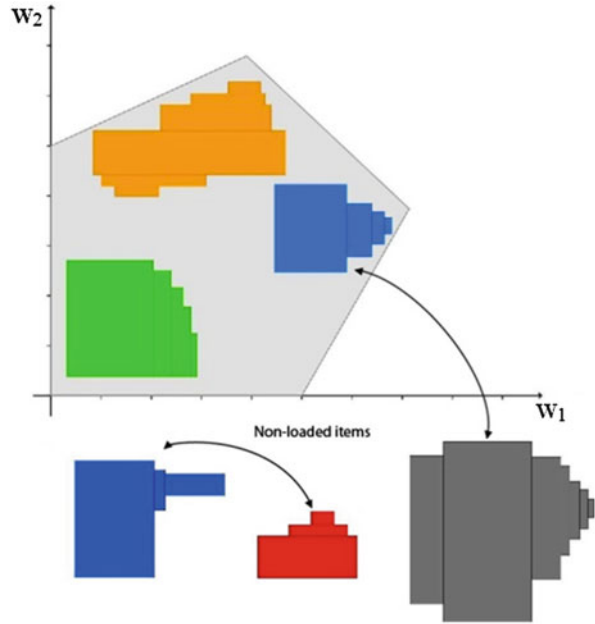
The task of this module is that of obtaining an *integer-feasible* solution to the general MIP model (with possible additional conditions, see Sects. 2.1 and 2.3) by imposing an *abstract configuration*. The *non-intersection* constraints (2.5a) and (2.5b), corresponding to the *relative positions* of the *abstract configuration* imposed, are kept unaltered, with their associated variables σ . All the remaining ones are instead eliminated. Inequalities (2.6) are reduced, for all the variables σ involved, to the following:

$$\forall i, j \in I / i < j, \forall h \in C_i, \forall k \in C_j \quad \sigma_{\beta h k i j}^{+/-} \geq \chi_i + \chi_j - 1. \quad (4.4)$$

All other constraints and variables of the general MIP model (including possible additional conditions) are kept and the *objective* function (2.7) still has the purpose of maximizing either the overall loaded volume or mass.

As an interesting alternative, the non-restrictive reformulation (Sect. 3.1.3), with all the relevant constraints, inclusive of possible additional conditions, and the *objective* function (3.10) could be used. The imposition of the *abstract configurations* would be carried out as explained above, involving the appropriate constraints.

Fig. 4.4 Item-exchange rationale (two-dimensional representation)



Remark 4.11 It can immediately be observed that because of constraints (4.4), the integrality condition on the variables σ may be dropped, so that they can simply be considered as continuous in the interval $[0,1]$. Furthermore, for each pair of components, one and only one of the non-intersection constraints (2.5a) and (2.5b) is still present, leading to a dramatic reduction of the original model dimension.

4.3.3.2 Item Exchange

This module is aimed at perturbing and tentatively improving the *abstract configuration* (referred to the whole set of items), corresponding to the current *packing* solution. An index permutation p among the set of items (and consequently in all the *relative position* constraints) is thus executed: $\forall i \in I, i \rightarrow p(i)$. In such a way, the overall effect consists of exchanging some of their relative positions, within the given *abstract configuration*, providing a new one.

In order to carry out (at least supposedly) promising exchanges, the following heuristic rationale, open to different possible versions, is proposed. It is sketched in Fig. 4.4. Picked items are exchanged with bigger non-picked items (or with items with bigger mass, if this corresponds to the optimization criterion chosen). Non-loaded items can also be exchanged. The way the above exchanges are implemented determines the specific strategy followed.

Remark 4.12 The item-exchange module performs actions likely to be advantageous in terms of loaded volume (or mass), just by performing permutations. It does

not take into account (directly) any constraints of the general MIP model that are, instead, contemplated by the packing module and (partially) by the hole-filling one.

Remark 4.13 Depending on the strategy adopted, this module, even by exchanging a limited number of items, can accomplish either a ‘weak’ or a ‘strong’ perturbation of the current abstract configuration. When a ‘weak’ perturbation strategy is executed, the exchanged items are not too different (in terms of volume and/or mass) from each other. They are, on the contrary, quite different, when a ‘strong’ perturbation strategy is chosen. When a ‘weak’ one is followed, the new abstract configuration remains ‘close’ to the previous, and the same is expected for the corresponding solution. This way, the general MIP model constraints (and the additional ones, when present) are therefore indirectly considered, through the ‘neighbourhood’ with the previous solutions .

Remark 4.14 If not all the exchanges carried out, in the new abstract configuration, between selected and nonselected items are feasible, the packing module is forced to actuate some rejections. It is, however, possible to avoid this inconvenience. Indeed, let us consider, for instance, the potential exchange of the picked item i with the non-picked one i' . It would be sufficient to duplicate, in the current packing MIP model, for item i' , all relative position constraints corresponding to i and pose the further condition: $\chi_i + \chi_{i'} = 1$ (updating, subsequently, the abstract configuration, on the basis of the obtained solution). This way the relevant exchange would not be imposed (preventing the possible consequence of rejection of both items).

4.3.3.3 Hole Filling

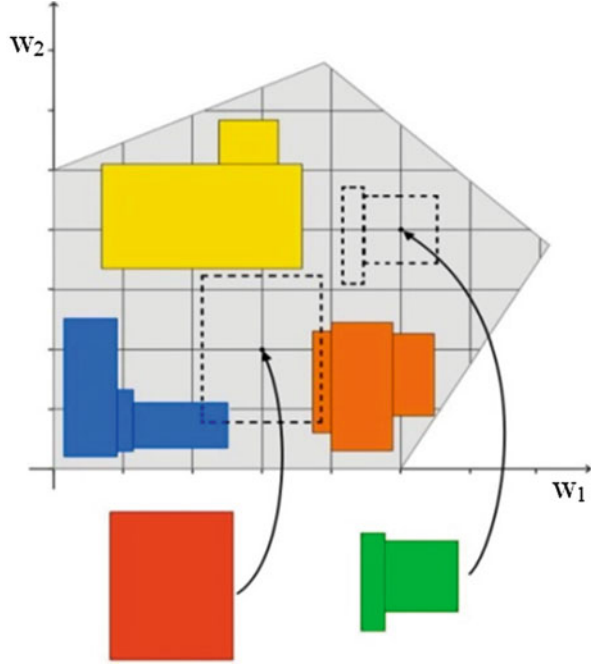
Also this module is aimed at perturbing the *packing* module (current) solution. Empty spaces are exploited by tentatively adding items extracted from the set \hat{I}_E of the currently excluded ones. This should produce an improved approximate solution (better in terms of volume or mass loaded, depending on the optimization criterion chosen, but with possible intersections) and an expectantly improved subsequent *abstract configuration*. To this purpose, the *packing* module current solution is ‘immersed’ into a grid domain, giving rise to a set \hat{N}_G of non-covered grid nodes; see Fig. 4.5.

The basic idea of the *hole-filling* module is that of selecting a subset $\hat{N}'_G \subset \hat{N}_G$ and one $\hat{I}'_E \subset \hat{I}_E$, of the currently excluded items, potentially associable to the chosen non-covered nodes (assuming $|\hat{I}'_E| \leq |\hat{N}'_G|$). This is aimed at obtaining (even if in an approximate way, i.e. with possible overlap) more loaded volume (or mass).

Three sub-steps are then considered hereinafter:

- *Non-covered node selection*
- *Excluded item selection*
- *Overall overlap minimization*

Fig. 4.5 *Hole-filling* rationale (two-dimensional representation)



The first and the second are skipped immediately, whenever the relevant sets have the desired cardinality.

4.3.3.4 Non-covered Node Selection

The following simple MIP model serves the scope of this sub-step. Denoting by ν the index of the generic non-covered node of \hat{N}_G , the binary variable ζ_ν is introduced, with the meaning $\zeta_\nu = 1$ if the corresponding (non-covered) node is selected; $\zeta_\nu = 0$ otherwise.

The conditions below are then posed:

$$\sum_{\nu \in \hat{N}_G} \zeta_\nu = |\hat{N}'_G|, \tag{4.5}$$

$$\forall \nu, \nu' \in \hat{N}_G / \nu < \nu' \quad \tilde{\zeta}_{\nu\nu'} \leq \zeta_\nu, \tag{4.6a}$$

$$\forall \nu, \nu' \in \hat{N}_G / \nu < \nu' \quad \tilde{\zeta}_{\nu\nu'} \leq \zeta_{\nu'}, \tag{4.6b}$$

where $\tilde{\zeta}_{\nu\nu'} \in [0, 1]$. The *objective* function below selects the (non-covered) nodes maximizing the overall relative distance:

$$\max \sum_{\substack{\beta \in B, \\ \nu, \nu' \in \hat{N}_G / \\ \nu < \nu'}} \left(W_{G\beta\nu} - W_{G\beta\nu'} \right)^2 \tilde{\xi}_{\nu\nu'}. \quad (4.7)$$

Here $W_{G\beta\nu}$ are the coordinates of the grid nodes. It is understood that alternative selection criteria could be chosen. This step provides, as outcome, the subset \hat{N}'_G of the selected non-covered nodes.

4.3.3.5 Excluded Item Selection

In this sub-step, $\hat{N}'_{Gi} \subseteq \hat{N}'_G$ denotes for each item $i \in \hat{I}_E$, the set of selected nodes that allow its placement inside the domain, for at least one orientation θ_{oi} . The following binary variables are introduced with the meaning $\xi_{i\nu} = 1$ if the excluded item i is associated to the node ν ; $\xi_{i\nu} = 0$ otherwise. The following equations guarantee that at most one item $i \in \hat{I}_E$ is allocated to the same node:

$$\forall i \in \hat{I}_E, \forall \nu \in \hat{N}'_G \quad \sum_{\substack{i \in \hat{I}_E / \\ \nu \in \hat{N}'_{Gi}}} \xi_{i\nu} \leq 1. \quad (4.8)$$

The following *objective* function substitutes (2.7) and maximizes the total volume (or mass) of the excluded items, associated to the grid nodes:

$$\max \sum_{\substack{i \in \hat{I}_E / \\ \nu \in \hat{N}'_{Gi}}} K_i \xi_{i\nu}. \quad (4.9)$$

The outcome of this sub-step determines the set of the items to add.

4.3.3.6 Overall Overlap Minimization

To perform this step, either the first or second linear reformulation (Sects. 3.1.1 and 3.1.2) is adopted. For the items already loaded, in the current *packing* module solution, the corresponding *abstract configuration* is imposed. All the *non-intersection* constraints involving the items selected to be added are, instead, generated. In such a way, the *objective* function minimizes the overall overlap. When balancing conditions are present, the relevant constraints are taken into account, together with the following equations:

$$\forall \beta \in B, \forall i \in \hat{I}_E \quad w_{\beta i}^* = \sum_{\nu \in \hat{N}_{Gi}} W_{G\beta\nu} \xi_{i\nu} \quad (4.10)$$

(where $w_{\beta i}^*$ are the centre of mass coordinates of the new items).

4.3.3.7 General Methodology Background and Alternative Approach

It is understood (see Fig. 4.3) that the *item-exchange* and *hole-filling* modules can be activated in various sequences, following different strategies. The *packing* MIP model is solved, each time, by adopting a *branch-and-bound*. Throughout this process, the binary variables χ , σ and θ are handled sequentially, by groups of items, prioritized by volume (or mass). A *depth-first* strategy is followed, during which subsets of binary variables can temporarily be fixed. A lower bound *cutoff* is set, on the basis of the best-so-far solution, and part of the items, previously picked, can be imposed, following a *greedy* search approach. If a satisfactory solution is found, it is taken as the ultimate one and the whole process ends. Otherwise, the best-so-far solution is stored and the process continues, until a (previously stated) stopping rule intervenes.

It should be further observed that the heuristic process discussed in this section, essentially, reproduces an overall (*delayed*) *column generation* philosophy (consult, e.g. the topical entry of INFORMS Computing Society 2013). The *packing* MIP mode, indeed, at each step, contemplates only a limited subset of variables σ , corresponding to the current *abstract configuration* imposed. This model thus plays the *master's* role, in a *column generation* framework. The generation of 'good' *abstract configurations*, and the corresponding selection of the variables σ , instead, represents the *pricing* phase, carried out by a heuristic process.

Before concluding this section, it is worth noticing that quite a promising alternative, compliant with the overall methodology adopted, consists of utilizing the non-restrictive reformulation of Sect. 3.1.3. This can substitute, tout court, both the *item-exchange* and *hole-filling* modules.

As far as the first is concerned, a number of items are still selected as candidates for the exchanges, as discussed above. Afterwards, all the *relative position* constraints, corresponding to the current *abstract configuration*, with at least one candidate, are substituted with the full set of *non-intersection* inequalities (2.6), (3.5a), (3.5b) and (3.9). The MIP model of the non-restrictive reformulation is hence adopted, keeping, for all the remaining items, the imposed *relative position* constraints. The objective function hence aims at carrying out the advantageous exchanges, improving, if successful, the current solution. In such a case, a new *abstract configuration* is directly generated. This is in general a larger perturbation of the current one, with respect to that obtainable by the *item-exchange* module.

As it is easily gathered, similar considerations hold when the *hole-filling* module is considered. In this case, all the *relative position* constraints corresponding to the items already loaded are kept. Those relevant to the items selected to be supposedly added are, instead, again substituted by the full sets of *non-intersection* constraints. The MIP

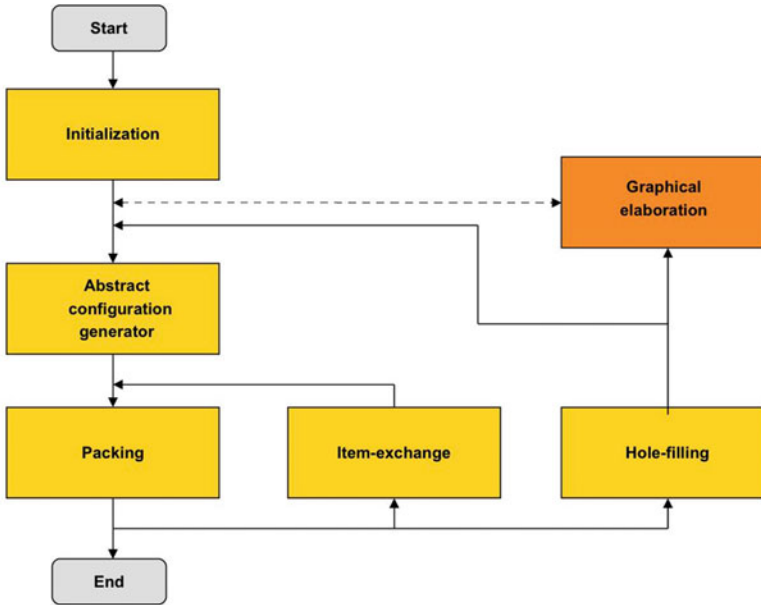


Fig. 4.6 Graphic-numerical interaction

model of the non-restrictive formulation tries and exploits (in a continuous mode) the volume still available and no discretization of the domain is needed any longer. If an improved solution is thus obtained, an upgraded *abstract configuration* is available.

4.3.4 Interaction with the Solution Process

As shown in Sect. 4.2, *abstract configurations* can easily be extracted by approximate solutions of the original problem. These are provided by the *initialization* and *hole-filling* modules. Any alternative process, however, up to yielding ‘good’ approximate solutions, could be activated as well. If, for example, a solution, of an instance similar to the specific one to sort out, is available, then it could be utilized to generate a first *abstract configuration*, skipping, directly, the *initialization* step. And similar opportunities could come up during the whole solution process, whenever a new *abstract configuration* has to be generated.

This suggests a sort of parallelization of human and computational capabilities by means of a two-way interface between the optimizer and a graphic system. The relevant approach can become quite effective in practice. Intermediate outcomes can, indeed, be visualized, time after time, and the natural human skills, up to managing (even very tricky) three-dimensional jobs by actually ‘seeing’ the objects involved, profitably exploited. This cannot only speed up the whole search for a satisfactory solution, but it also allows the extemporaneous introduction of further conditions hard to formulate explicitly in the model (for instance, ergonomic conditions). Figure 4.6 illustrates the two-way interface rationale.