

Computer tool for maximizing the placement of congruent polyhedra

by M. GASPAR^a, N. ALVES^a, A. MATEUS^a AND N. MARTINS-FERREIRA^a

^a Centre for Rapid and Sustainable Product Development,
Polytechnic Institute of Leiria, Marinha Grande, Portugal

Abstract: Given multiple identical polyhedral objects and a parallelepiped container, how should one place the objects so that the largest number fits inside the container? This simple question is important in many applications, yet the answer is elusive. In fact, we know of no published solution for this very general formulation. Still, in many circumstances, further restrictions apply, resulting in a large number of variations requiring different algorithmic strategies. This paper is the continuation of [12] and focus on the fundamental concepts and tools that are used for this kind of problem, such as the no-fit polygon. We also present some of its many variations, giving in particular one that applies to the stereolithographic rapid prototyping technology.

Keywords: Rapid Prototyping, Stereolithography, Containment, Packing, Optimization.

1 Introduction

The problem of maximizing the number of items that fit inside a container is one of a family of problems known as Cutting and Packing (C&P), most of which are notoriously hard. For an overview of this family of problems, see [4]. These problems include for instance the cutting of large sheets or rolls of paper to produce a given number of specific size sheets, with minimal waste; or the order and positions for placing different-sized boxes in a container.

The particular problem we will address is that of placing the maximum number of objects in the work area of an additive rapid manufacturing or rapid prototyping (RP) machine. This is most relevant for those RP techniques where a whole layer is processed at a time, as is the case in stereolithography. When using extrusion-based processes, we are usually interested in minimizing the total path length, and thus software for those machines will try to minimize the total area occupied by a user-specified set of parts. In stereolithography, on the contrary, since the irradiation of a layer is done by projection, it takes an approximately constant time and energy to process each layer, regardless of the fraction of the work area that is occupied by the fabricated parts. So, placing more parts in the work area directly translates to time- and energy-efficiency.

Several methods have been proposed for packing multiple distinct parts in a container, either minimizing the length of the container or maximizing the total number of parts. Most of those methods have some of the following characteristics: based on heuristic, iterative, stochastic; and most are formulated for two dimensional problems or for a restricted set of part geometries (for instance, parallelepiped boxes). For large production runs, trial-and-error, progressive optimization, or processing-intensive algorithms may be useful. But for the small

or one-time series that are typical of rapid prototyping/rapid manufacturing, a fast method for obtaining near optimal layouts is desirable.

Although this application does not require it, we will consider only identical, congruent objects placed in a regular lattice. Also, as it is the most common case, only parallelepiped working areas will be considered.

This problem involves two limit-problems, according to the relative dimensions of the container and the contained objects: if the dimensions of the container are much larger than those of the objects, the solution is mostly dependent on the density of the placement; conversely, if the dimensions are similar, the solution depends more on the intersection of the objects with the boundary of the container. In the first case we have a packing problem, whereas in the second case we have a containment problem.

Typically, the objects to fabricate cannot be stacked vertically, since they must be supported against gravity. Further, as there is a privileged direction (the direction of slicing), the orientation of the parts may be restricted in one axis, for proper surface finishing or mechanical properties. These two restrictions are important, as we shall see, in that they allow us to reduce this three-dimensional packing problem to a two-dimensional equivalent.

In this paper we will focus on the fundamental concepts and tools used for this kind of problem, such as the no-fit polygon. These concepts will be illustrated by means of an interactive software application we developed in Matlab. We briefly present two previously published methods for finding optimal or nearly-optimal solutions to the packing and containment problems in two dimensions [1, 5], the limitations and advantages of each method, and how they may be applied to this problem. Finally we discuss how the interactive application developed can be useful in exploring the application of similar techniques to a few other three-dimensional problems.

2 Description of the method

The solution to the problem stated in the introduction involves several conceptually distinct optimization problems; this separation is helpful in understanding the problem, even if not all sub-problems can be treated independently. The following subsections deal with each of the steps presented below:

1. **Reduction of the three-dimensional problem to a two-dimensional one.** In this section we will analyze the constraints under which it is possible to apply the algorithms for the planar packing for problems formulated in three-dimensional space.
2. **Densest linear packing of polygons.** In this section we will present the concept of no-fit polygon, how it is computed, and how it captures the non-overlapping constraint in one dimension.
3. **Finding feasible solutions to dense regular packing of polygons.** In this section we will again use the no-fit polygon, and show how it can be extended to find non-overlapping tiling of the plane.
4. **Determination of the densest tiling.** This section mainly describes the method proposed by Stoyan and Patsuk [1] to identify, among all non-overlapping tiling, those of maximum density.
5. **The Containment Problem.** In this section, for further development, we give some directions for determining the translation and rotation of a (densest) periodic tiling that minimizes the number of lattice cells intersecting the boundary of the container, on the one hand, and if there exists a non-optimal lattice that may produce a better overall solution, on the other.

2.1 Reduction of the three-dimensional problem to a two-dimensional one

The condition of no vertical overlap means that for some given orientation of a polyhedron P , we need only consider the packing of its projection on the horizontal plane (we also use P to refer to that polygonal projection). Thus, if we additionally restrict the rotation of P only to the vertical axis, the problem is effectively reduced to the two-dimensional case. The same reasoning can be applied to the containment problem.

Although rapid fabrication is sometimes used for the production of pre-assembled multi-part structures, we will restrict our analysis to the single part case, i.e., where the interior of the polyhedral surface is simply connected. This implies that the projection along any direction will have a simply connected interior. Note that its boundary may still not be a simple polygon. There may be 'holes' in the polygon, but clearly the polygon will not fit inside

its own holes; so, in fact, we may consider only the outer boundary of the polygon.

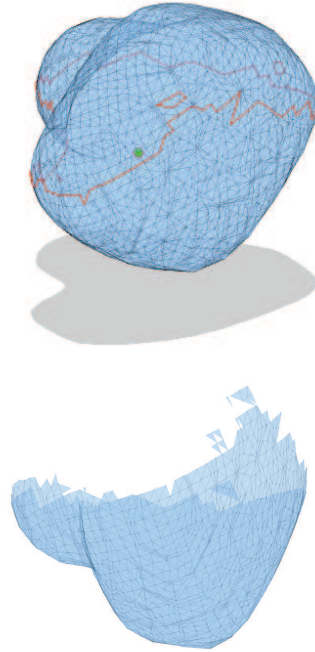


FIGURE 1: A polyhedron with a polygonal contour separating facets with upward and downward normals. The union of the contour projections is represented as the shadowed area. Downward oriented facets are displayed in the separated picture.

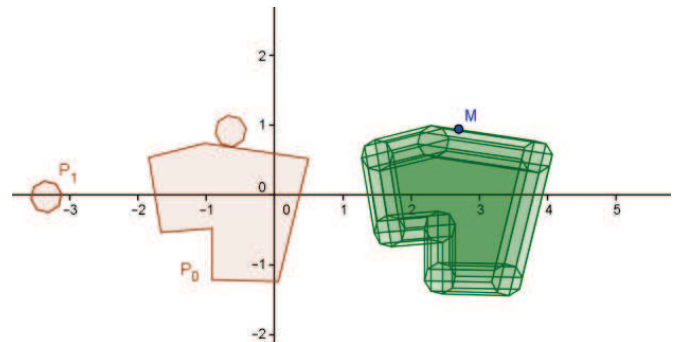


FIGURE 2: Dilation of a polygon by a regular octagon (approximation of circumference) equate to enforcing a separation between parts.

Obtaining the polygonal boundary of the projection can be done as follows: find the sets of edges that separate facets with upward and downward normal; chain those edges to form closed oriented contours; finally, obtain the union of the regions bounded by those oriented contours projected into the plane, as illustrated in the upper part of Figure 1.

In many cases it may be required to have a minimum separation between adjacent parts. This can be enforced by replacing the part with its dilation by sphere of proper

radius. But since we will only deal with the projection, an easier approach is to apply, say dilation, to the projection. This procedure is illustrated in Figure 2.

The exact shape of the dilation of a polygon by a circumference will not be polygonal, since it will contain arcs of circumference. Still, as the desired gap will usually be much smaller than the typical dimensions of the part, it can be approximated by a regular polygon. The dilation of a polygon by another polygon can be computed by using the same concept of no-fit polygon that we will use for defining the non-overlapping condition, which will be introduced in the next section.

2.2 Densest linear packing of polygons

In order to find the optimal lattice packing, we must be able to efficiently determine when two objects overlap in any given direction. This is done via the concept of no-fit polygon (NFP) or the closely related phi-function 0-level [2].

The no-fit polygon represents the set of translations of a polygon P_1 such that P_1 and another polygon P_0 just touch; it can also be viewed as the boundary of the region defined by the points $P=(r;\theta)$ such that the translations of polygon P_1 by OP does intersect polygon P_0 . As the name implies, this boundary is polygonal. We will denote this as $NFP(P_0, P_1)$.

It is important to note that the NFP may be non-convex and even non-simple – i.e., composed of multiple disjoint polygons. The meaning of a non-simple NFP is that some admissible translations might require ‘lifting’ one of the polygons (see for instance figure 3 below). For some applications this positions might not be acceptable, as the two pieces become interlocked.

The NFP may also contain zero-area components: an isolated point or loops of parallel overlapping edges. Those correspond to tight fit of the two polygons such that they can not move (isolated point) or can only move in a linear path. When we consider only the boundary of the NFP without those degenerate cases, it is called the ‘regularized NFP’ [11].

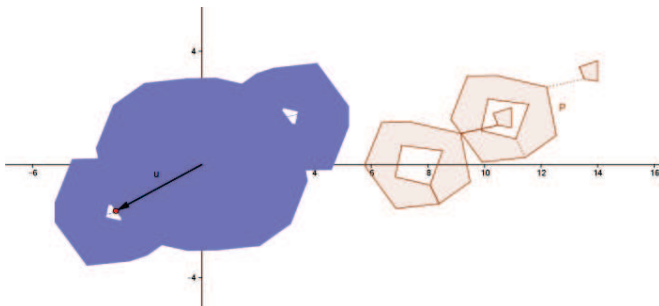


FIGURE 3: Interlocking and non-simple NFP. The holes in the NFP (on the left) represent non-overlapping translations of polygon P that can not be achieved without crossing P .

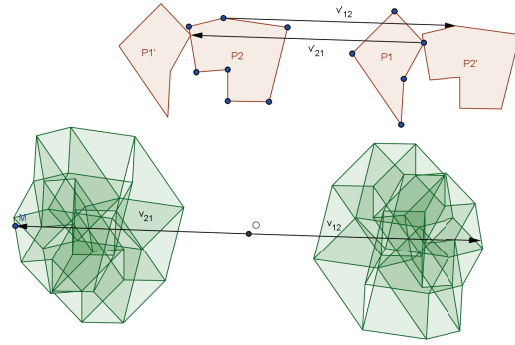


FIGURE 4: The NFP of polygons P_1 with P_2 , and P_2 with P_1 , and how they relate to the translations of the second polygon that do not overlap the first one. Point M on the no-fit polygon of P_2 with P_1 (bottom left) corresponds to the translation of P_1 by $v_{21} = OM$ such that P_1' touches P_2 but doesn't intersect it. Moreover, considering a point in P_2' and sliding it around P_1 will trace the outer contour of the $NFP(P_1, P_2)$ (bottom right).

A mechanistic interpretation of the NFP, is that we bring the two polygonal regions to contact and slide one of them (P_1) along the boundary of the other, tracing the position of a fixed point in P_1 (see for instance Fig. 2 or Fig 4). This clearly shows that the boundary of the NFP is formed by the edges of the two original polygons. Indeed, if both polygons are convex, the sliding movement is along the full extent of each edge of both polygons, whereas if any of them is non-convex or non-simple, sliding traverses only part of some edges. Note that if any of the polygons is non-convex, not all edges will be necessarily present in the NFP.

The NFP is defined for two polygons, but for our application we shall consider the NFP of a polygon with itself, which we will denote by $NFP(P) = NFP(P, P)$. In this case, the NFP will obviously have central symmetry, since $NFP(P_0, P_1) = -NFP(P_1, P_0)$. Additionally, if P is convex, $NFP(P)$ is also convex; and if P is a star polygon, the $NFP(P)$ is also a star polygon. Further properties of the NFP can be found in [2].

The NFP can also be used to enforce the minimal gap between adjacent parts, as mentioned in previous section. Let us consider only a dilation of a polygon by a circle. If we consider the NFP of P with a circle of radius R , by the orbiting or sliding interpretation, using the center of the circle as the reference point, we can see that each point of the NFP will be at a distance at least R from P . So, if we take this NFP as the new polygon P' , the condition that $P' + v$ touches P' implies that any point in $P + v$ is at a distance of at least $2R$ of a point in P .

2.3 Methods for computing the NFP

Efficient methods for computing the NFP of two polygons are mainly based on two approaches: the edge precedence and the Minkowski sum [2].

The first technique is simple and efficient, but only for the convex-convex or simple-convex cases. It is based on the sliding interpretation described above, and mainly uses the ordering of the slope of edges of both P_1 and the reflection of P_2 about the origin; for convex-convex case, the NFP is obtained by chaining together all edges in that order.

The second method, on the other hand, has been effectively used for arbitrary polygons, at the cost of computational complexity. The Minkowski sum, represented by \oplus , is defined for two sets of points S_a and S_b as the set of points $S = \{a + b \mid a \in S_a, b \in S_b\}$. If we consider S_a and S_b as the interior of P_a and P_b , we can define the Minkowski sum of two oriented polygons $P_a \oplus P_b$ as the boundary of $S_a \oplus S_b$. If we consider two edges $e_a = AA'$ and $e_b = BB'$, we may define their Minkowski sum as the parallelogram with edges e_{a+B} , e_{b+A} , $e_{a+B'}$ and $e_{b+A'}$. For instance, in figure 8b, the parallelogram Tij is the Minkowsky sum of edges e_i and e_j .

The formal analysis of how the Minkowski sum is related to the NFP was done by Stoyan and Ponomarenko [9], who related the NFP to the boundary of $\text{int}(P_a) \oplus (\text{int}(-P_b))$. Ghosh showed that the NFP is determined only by the boundaries of both polygons, and developed methods for computing the NFP as a combination of the edges of the polygons, embodied in the edge precedence approach (see [10] and its references).

A full description of a general method for computing the NFP of two arbitrary polygons can be found in Ben-nell and Song [2]. It is based on the fact that the NFP of a convex polygon with a simple polygon can be easily computed by the edge precedence method; thus, one of the polygons may be transformed in a convex polygon by replacing concave edges, for computing an approximate NFP; then the 'fake' edges are replaced by the original concave ones, and finally the real boundary of the forming the NFP is obtained by appropriate clipping of the edges. This is a fairly complex algorithm, and computationally expensive.

For this application, only the NFP of a polygon with itself is needed. Also, as stated previously, we will only consider the cases where the polygon is simple. As such, in our work we use a simplified version of the Minkowski sum technique, which is easier to describe and implement. The procedure is based on two operations: the union of two polygonal regions and the removal of cyclic components. For methods that can be used to perform these operations see [8].

S_1 and S_2 being the sets of vertices of each polygon, we may define the Minkowski sum of P_1 and P_2 as the union of all translations of P_2 by \overrightarrow{OV} , for each V in S_1 .

$$P_1 \oplus P_2 = \bigcup_{V \in S_1} (P_2 + \overrightarrow{OV})$$

With this definition in mind, the first step of our procedure consists in computing

$$P_{1,2} = (P_1 \oplus (-P_2)) \cup ((-P_1) \oplus P_2)$$

(where $-P$ represents the reflection of P about the origin).

For a single polygon P , the procedure is further simplified, since

$$\begin{aligned} P_{12} &= (P \oplus (-P)) \cup ((-P) \oplus P) \\ &= (P \oplus (-P)) \cup -(P \oplus (-P)) \end{aligned}$$

Note that the obtained polygon $P_{1,2}$ is not the NFP, since it will possibly contain some additional edges, forming loops or cycles connected to the external boundary of $P_{1,2}$ (see figure 5b). Those cycles have a particular meaning: they represent the positions where some edges of one polygon cross an even but non-zero number of edges of the other. This implies that edge begins and ends on the outside of the other polygon, while still intersecting it; a translation vector with endpoint in the region limited by such loop will leave all vertices of P' outside of P .

In order to obtain the true NFP, regardless of whether we are computing the NFP for two or one polygon, we must remove the cycles. But if we are not interested in interlocking packings, we may as well remove all interior regions, that is, keep only the outer boundary of $P_{1,2}$.

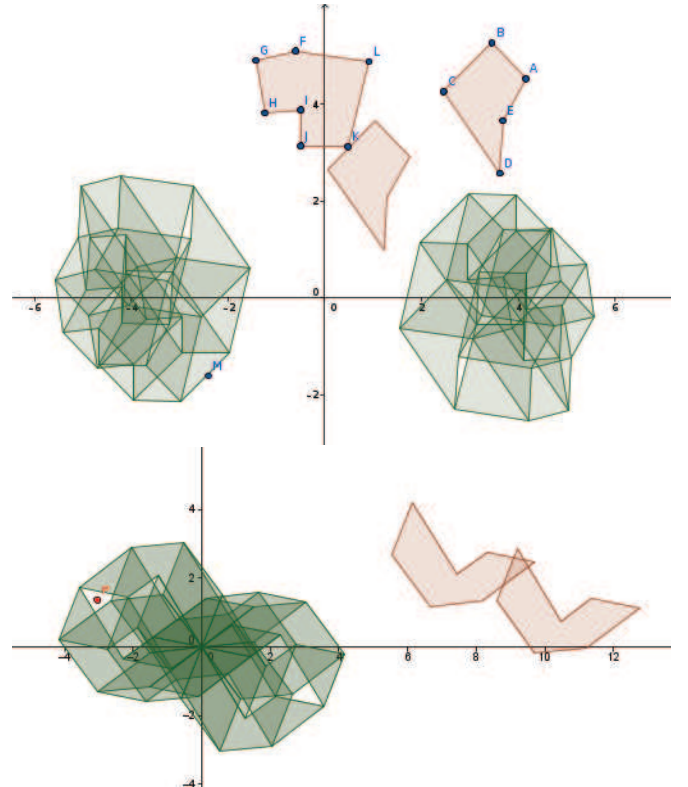


FIGURE 5: Computing the NFP of a) two polygons or b) one single polygon with itself. In b) the point P (red) lies in a 'hole' of P_{12} ; the corresponding translation, while leaving all vertices of the translated polygon outside of the original, still leads to intersection.

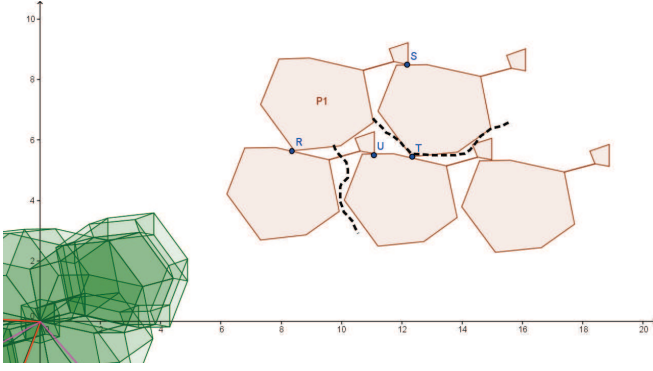


FIGURE 6: Conditions for a dense packing. If P is simply connected, and it touches $P + u$ and $P + v$, and does not intersect $P + (u \pm v)$, then it can not touch any other neighbor, and (u, v) generate a packing.

2.4 Finding feasible solutions to regular planar packing of polygons

Given the no-fit polygon (NFP) of polygon P with itself, we seek to determine all lattice bases, defined by vectors \vec{v}_1, \vec{v}_2 that produce non-intersecting tiling of P . This seems to imply that any linear combination of \vec{v}_1 and \vec{v}_2 with integer coefficients must be considered for possible intersection. We will see that for our purpose we need only consider a subset of regular packings, called dense packings. A dense packing is defined as a packing where adjacent polygons touch in at least two directions.

By considering the paths connecting two copies of P , we can see that the no-overlap conditions can be simplified for dense packings, provided that $\text{int}(P)$ is simply connected. Indeed, in two dimensions, provided that two immediate neighbor copies of P , i.e., with $(n, m) \in \{-1, 0, 1\}^2$, touch P , there is no need to consider other possible intersections. Figure 6 shows that the points $A, B, A' = A - v_1$ and $B' = B - v_2$ where P touches its immediate neighbors in a dense packing must be connected by a part of the frontier P ; since $B' + v_2 = B$, the immediate neighbors form a closed path, so that there is no possible path connecting P to higher order neighbors without entering a first-order neighbor.

The dashed lines in Figure 6 illustrate why this fails if $\text{int}(P)$ is not simply connected (the downward path), or if the packing is not dense.

Notice that the same reasoning does not apply in three dimensions. Since in that case the touch points do not ensure a closed surface, the higher order neighbors are not isolated.

From the above, we can use the NFP to restrict our search to only the lattice packings that are possibly optimal as follows: the bases formed by vectors \vec{v}_1, \vec{v}_2 with origin in the center of the NFP, endpoint on its boundary, and such that their sum and difference are not in the interior of the NFP:

$$\begin{aligned} \vec{v}_1 &= OP, \vec{v}_2 = OQ, \text{ with } P, Q \in \partial NFP \\ \vec{v}_+ &= \vec{v}_1 + \vec{v}_2, \vec{v}_- = \vec{v}_1 - \vec{v}_2, \text{ with } \{\vec{v}_+, \vec{v}_-\} \notin \text{int}(NFP) \end{aligned}$$

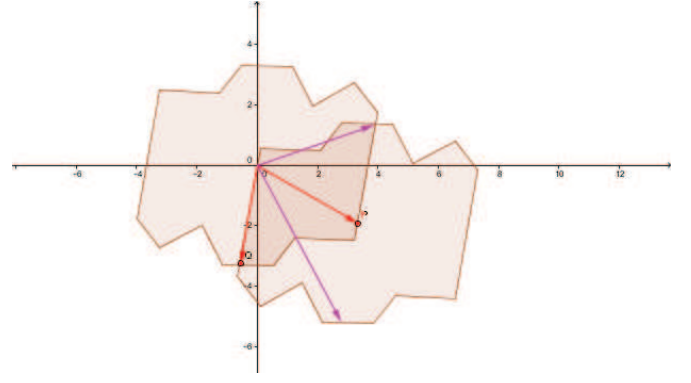


FIGURE 7: There is always a pair of vectors (u, v) such that $u+v$ lies on the NFP.

Note that given one vector ($v=OP$) lying on the boundary of the NFP, there is always at least one pair of vectors v_1 and v_2 also on the boundary of the NFP, such that their sum is $v=v_1+v_2$ (see figure 7).

2.5 Determination of the densest packing

If the dimensions of the polygon P are much smaller than the dimensions of the container, the number of parts that can be placed on that container will depend more on maximizing the density of the packing than on minimizing the intersections with the container. In fact, the number of intersected parts will be broadly proportional to the perimeter of the container, whereas the number of non-intersected parts will be related to the ratio of the areas of the polygon and the container.

The density of a packing generated by vectors v_1 and v_2 is defined as the ratio of the area of the polygon to the area of the lattice cell (area of a parallelogram with sides v_1 and v_2); since the former is fixed, in order to maximize the density we must find the minimum of the area for the lattice cell (see Figure 8(a)).

Stoyan and Patsuk [1] prove that for any given non-dense packing, there exists always a dense packing (as defined in previous section) with no-lesser density. That means that we can restrict our search for the densest packing to only the cases where the two vectors lie on the NFP.

We present next an outline of the concepts behind the approach proposed in [1].

We start by considering the area of a parallelogram C , formed by two vectors v_1 and v_2 , with origin in the center of the NFP and endpoints on two edges, e_i and e_j (see figure 3). The area of C is given by the magnitude of the cross product of v_1 and v_2 .

The sum $v_+ = v_1 + v_2$ for all combinations of positions of both vectors define another parallelogram T_{ij} , whose sides are the edges e_i and e_j , translated by v_2 and v_1 , respectively. It can be defined also as the Minkowski sum of the two edges. The area of C decreases monotonically as each vector's endpoint moves along each edge, so it will have a minimum at a vertex of T_{ij} , that is, when both

vectors of the base are on a vertex of the corresponding edge. If all four combinations of vertices for a pair of edges are admissible, then the minimal area for bases generated from that pair is among the values obtained for each combination.

On the other hand, the region T_{ij} also represents the positions of $v_1 + v_2$. For a base (v_1, v_2) to be admissible, $v_1 + v_2$ must not be inside the NFP. Let e_k be one edge touched by $v_1 + v_2$, for some v_1 on e_i and v_2 on e_j . It should be noted that, for a simple polygon, if $v_i + v_j$ touches the NFP, then $v_i - v_j$ will always be outside the NFP; thus, these configurations are always feasible (if P is non-simple, this does not hold, and we would need to consider both the intersection of $v_1 + v_2$ and $v_1 - v_2$). So, the admissible solutions involving edges e_i and e_j correspond to a sub-region E_{ij} of T_{ij} , the boundary of which is composed from whole or parts of edges of the NFP.

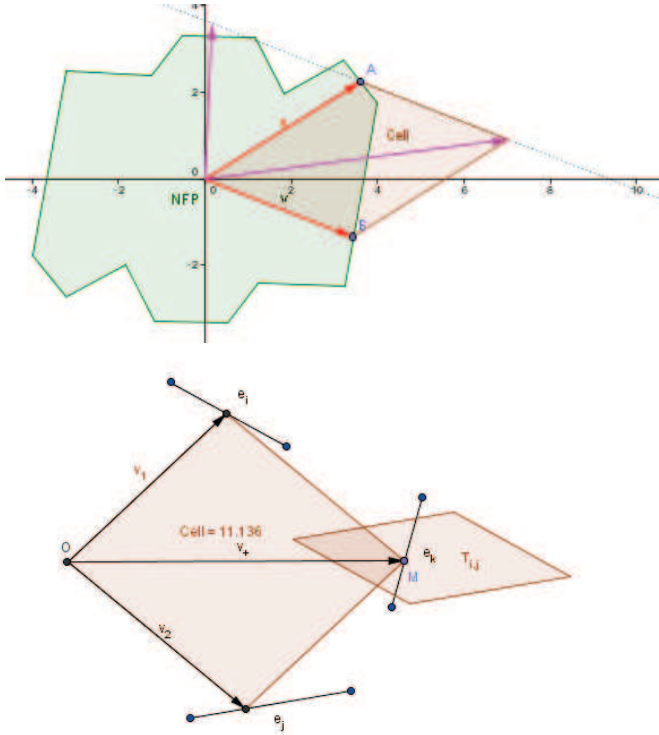


FIGURE 8: The possible lattices generated by vectors on a) a pair of edges, or b) three edges, are in correspondence with the set of points in T_{ij} or the intersection of e_k with T_{ij} , respectively.

The position of v_1 along e_i uniquely determines the positions of v_2 along e_j such that $v_1 + v_2$ touches e_k . We can thus easily find the position that minimizes the area for each combination (e_i, e_j, e_k) . If some edge of T_{ij} intersects the NFP at some edges e_k , an odd number of times, then at least one vertex of T_{ij} is not admissible. We can find the set of edges e_k that are partially or fully contained in T_{ij} . For each such edge, consider its supporting line; we want to find the minimal parallelogram touching three straight lines, at P_i, P_j , and P_k , with a fixed fourth

vertex. Let t_k be the parametric coordinate of a point along edge e_k ; the corresponding t_i and t_j are uniquely determined by t_k . If the parametric representations of P_i, P_j , and P_k , respectively t_i^*, t_j^* and t_k^* are all within $[0, 1]$, then t_i^* and t_j^* determine a local feasible minimum.

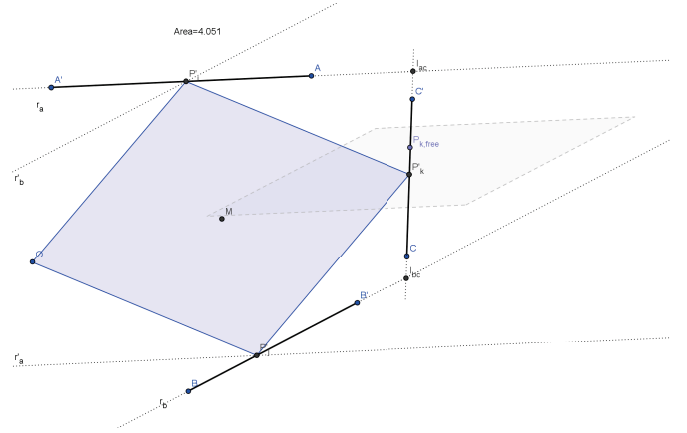


FIGURE 9: Geometric determination of the minimal parallelogram touching three edges of the NFP. P_k^* is the midpoint of segment $I_{ik}I_{jk}$. If all of P_i, P_j and P_k are inside the respective edges, the corresponding lattice generates a dense packing with maximum density for those edges.

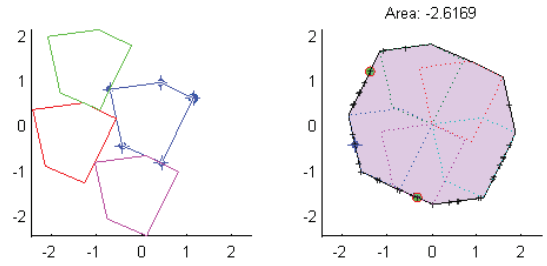


FIGURE 10: The plot on the right shows the NFP for the blue polygon on the left. On the NFP are plotted (+) the possible positions along each edge maximizing the lattice density. The maximal density lattice packing is also represented on the left plot, and its base vectors correspond to the green dots on the NFP.

Geometrically, this can be obtained by the following construction: given an origin O and three edges (e_i, e_j, e_k) , the point on edge k that gives the minimal area kissing parallelogram – parallelogram with one vertex on O and the other three on each given edge – is the midpoint of the intersections of the supporting lines for each of e_i and e_j with the supporting line of e_k (See Figure 9). Note that this is only true for edges with proper relative orientation; but it is always possible to take the proper oriented edges by replacing e_j by its symmetric $-e_j$, which also is part of the NFP, and generates the same lattice. Note also that the same procedure can be also applied to find the corresponding vertices in e_i or e_j , if we replace e_j (respectively e_i) by its symmetric; alternatively, if we take the midpoint M of OP_k , since it is the

center of symmetry of the parallelogram, the other two intersections can be obtained by intersecting the reflection of r_b about M with r_a .

The points (P_i, P_j, P_k) may or may not be all contained in the corresponding edges.

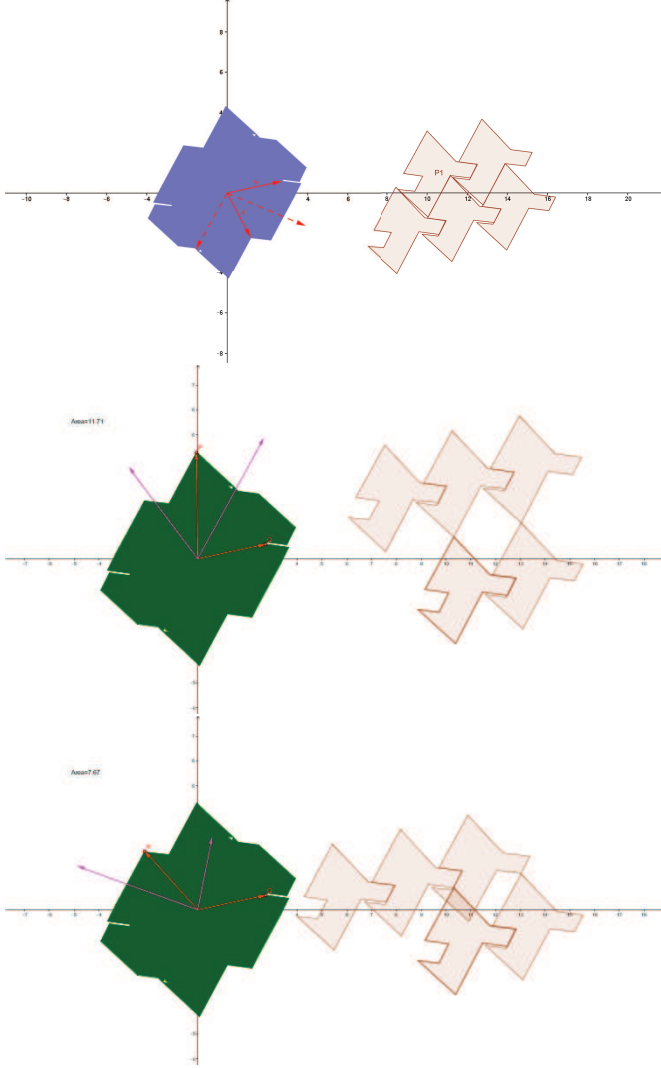


FIGURE 11: Optimal (a) and non-optimal (b,c) lattices.

The optimal solution can then be found by computing the area for all pairs of vertices of the NFP, and checking if each pair is admissible; computing the optimal position (if it is admissible) for each triple of edges, and the corresponding area; and searching for the minimum of the computed values.

2.6 The Containment Problem

Having determined a densest infinite packing of a polygon, we still have to consider if it leads to a maximal number of fully-contained copies for a given container size. In general, we have three translational degrees of freedom, plus the orientation, for the placement of the lattice in the container. In our specific application, however, the vertical translation is fixed by the requirement

that the objects be supported by the work plane. Similarly, the rotation of the lattice is restricted to the vertical axis, so that the preferred orientation is maintained.

Although this is a very important application case, there are very few published works on the packing problem which take into account the container.

If we are considering a bounded packing, we must take into account that the packing distance is in general smaller than the projection of the polygon perpendicular to the packing direction – that is, the polygon will extend beyond the 'unit' of the packing lattice. This 'protruding' part of the polygon can not however cross the bounds of the container, thus reducing the 'effective' container length available for the lattice. This is illustrated for a linear bounded packing in Figure 12.

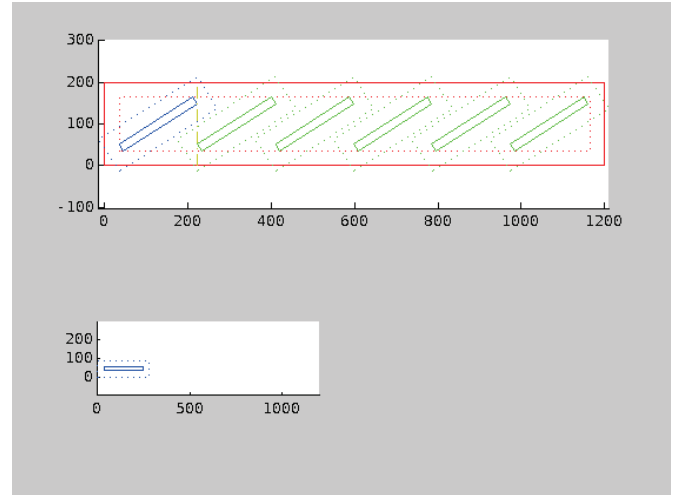


FIGURE 12: Linear bounded packing.

Although we have no closed form expression for the number of pieces contained in (or intersected by) a rectangular container as a function of the lattice position, we can establish upper and lower bounds. The upper bound is obtained by simply dividing the total area of the container by the area of the polygon. A possible lower bound reduces to the regular packing of spheres (or circles) that circumscribe the polyhedron (or polygon). This observation is useful in restricting the search-space for the optimal solution.

For a bounded packing, we will want to maximize the number of parts fully contained. Several heuristics may be applied. We may expect the densest unbounded packing to provide a near-optimal solution, and deal with both problems separately – finding a densest packing, and then finding the optimal placement of that packing relative to the container. We could also look for the densest rectangular packing, based on the assumption that a rectangular packing will be able to provide a better solution for the containment problem. Or we could consider the linear bounded packing, along one side of the container, followed by the packing of the obtained strip along the other direction.

In order to quickly produce near-optimal layouts, while retaining the ability to explore all of the problem space, including the three-dimensional cases, we developed an interactive application. The main window of this application is presented in Figure 13. After selecting an arbitrary 3D-part (described as an STL file) or parameterizing a parametric model (extruded part, swept part), the user can specify the vertical orientation and a margin around the part. The application then computes a densest lattice packing in the horizontal plane, which the user can freely modify to explore other dense or arbitrary (non-dense) packing. For each lattice selected, an optimal placement within the container is computed, which can also be manually adjusted, and the resulting number of parts is computed.

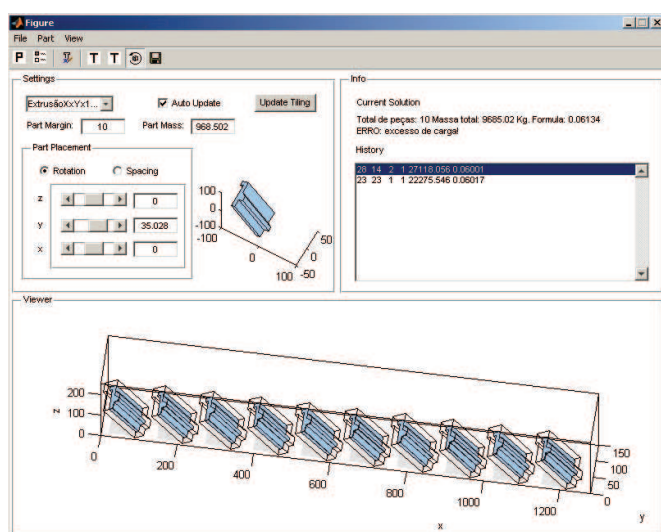


FIGURE 13: An interactive application for exploring feasible solutions to the dense regular planar packing of a polygon.

3 Conclusions

The published methods referenced above each have some limitations. The method by Stoyan and Patsuk [1] provides only a partial solution – limited to the densest lattice packing of a single polygon – but it gives an exact solution. On the other hand, the method proposed by Milenkovic [5] is very general, being able to deal with the placement of multiple different polygons. This generality comes at a cost: as he points out, his method only provides an approximation, albeit with a very tight bound, to the optimal solution.

The solution presented in this paper was specially designed for the particular case of maximizing the placement of congruent polyhedra in a parallelepiped container. It was obtained by the reduction, under suitable restrictions, to a two-dimensional case. The problem can then be solved by applying the method pre-

sented in [5]. Still, the performance and complexity of that method raises some issues. As an alternative to a fully automated solution, an interactive application was developed in which near-optimal solutions for the problem can be found with little effort. By an extensive analysis of the sub-problems we were able to combine known tools and heuristics allowing an efficient exploration in the problem-space. These tools may prove useful in finding solutions for related, less constrained, problems.

4 References

- [1] A method of optimal lattice packing of congruent oriented polygons in the plane, Y.G. Stoyan, V.N. Patsuk, *EJOR* 124 (2000) 204 – 216
- [2] A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums, J.A. Bennell, X. Song, *Computers & Operations Research* 35 (2008) 267 – 281
- [3] The geometry of nesting problems: A tutorial, J.A. Bennell, J.F. Oliveira, *EJOR* 184 (2008) 397–415
- [4] An improved typology of cutting and packing problems, G. Wascher et al., *EJOR* 183 (2007) 1109–1130
- [5] Densest translational lattice packing of non-convex polygons, V. Milenkovic, *Computational Geometry* 22 (2002) 205–222
- [6] Novas abordagens ao Posicionamento Periódico de Figuras Irregulares, M. T. Costa., PhD Thesis, Universidade do Porto, 2007.
- [7] Tools of mathematical modeling of arbitrary object packing problems, J. Bennell et al., *Ann Oper Res* (2010) 179: 343–368
- [8] A method for computing the symmetric difference of arbitrary regions in the plane, M.B. Gaspar and N. Martins-Ferreira (submitted)
- [9] Minkovsky sum and the hodograph of the dense allocation vector-function, Yu.G. Stoyan and L.D. Ponomarenko, *Reports of the Academy of Sciences of Ukrainian SSR, Ser.A(Russ)*, 1977, p.10
- [10] P. K. Ghosh, A unified computational framework for Minkowski operations, *Computers & Graphics*, Volume 17(4) (1993) 357–378.
- [11] A Simulated Annealing Based Algorithm with Collision Free Region for the Irregular Shape Packing Problem, A.K. Sat, T.C. Martins, M.S.G. Tsuzuki, preprints of the 18th IFAC World Congress, 2011.
- [12] Maximizing the placement of congruent polyhedra in a parallelepiped container, Gaspar, M. Belbut; Martins-Ferreira, N., *Numerical Analysis and Applied Mathematics ICNAAM 2012: International Conference of Numerical Analysis and Applied Mathematics. AIP Conference Proceedings, Volume 1479. AIP Conference Proceedings, Volume 1479, Issue 1, p.1588-1591*