Mathematical model and efficient algorithms for object packing problem

N. Chernov[1], Yu. Stoyan[2], and T. Romanova[2]

December 1, 2008

Abstract

The article is devoted to mathematical models and practical algorithms for solving the cutting and packing (C&P) problem. We review and further enhance the main tool of our studies – phi-functions. Those are constructed here for 2D and 3D objects (unlike other standard tools, such as no-fit polygons, which are restricted to the 2D geometry). We also demonstrate that in all realistic cases the phi-functions and their derivatives can be described by quite simple formulas without radicals and other complications. Lastly, a general solution strategy using the phi-functions is outlined and illustrated by several 2D and 3D examples.

Keywords: Mathematical modeling; Cutting and packing; Optimization; Phi-function; No-fit polygon

1 Introduction

The cutting and packing (C&P) problem is a part of computational geometry that has rich industrial applications in garment industry, sheet metal cutting, furniture making, shoe manufacturing, etc. The common task in these areas is to cut a certain set of figures of specified shapes and sizes from a given sheet

[1]Department of Mathematics, University of Alabama at Birmingham, AL 35294;
[2]Department of Mathematical Modeling, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Kharkov, Ukraine;
Email: chernov@math.uab.edu; sherom@kharkov.ua

(strip) of material (textile, wood, metal, etc.), see a tutorial [5] and references therein. To minimize waste one wants to cut figures as close to each other as possible; in other words one needs to design an optimal (compact) layout before the actual cutting.

This is a mathematical problem which can be formalized as follows: given a strip of fixed width W and infinite length, say $S = \{x \geq 0,\ 0 \leq y \leq W\}$, cut out n given figures from the rectangle $\{0 \leq x \leq L,\ 0 \leq y \leq W\} \subset S$ without overlaps, so that L takes its minimum value, see Figure 1.

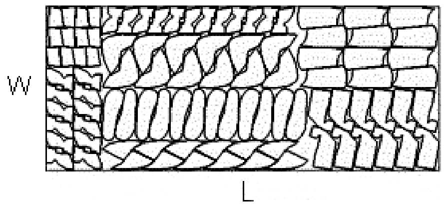

Figure 1: An example of a cutting problem.

In other applications, one needs to arrange a given set of objects within a certain area (say, shipment on a deck of a freight car or electronic components on a panel), and again one wants to minimize the use of space or maximize the number of objects.

Clearly these two types of problems – cutting and packing – are mathematically equivalent; they are known as the cutting and packing (C&P) problem (it is also called nesting problem, marker making, stock cutting, containment problem, etc). In some cases it involves additional restrictions on the minimal or maximal distance between certain objects or from the objects to the border of the container. The recent tutorial [5] summarizes the previous studies of the C&P problem and its history.

Many other applications involve 3D geometry: packing pills into a bottle, placing crates and barrels into a cargo compartment, 3D laser cutting, modeling of granular media and liquids, and radiosurgery treatment planning (just to name a few). Thus the C&P problem naturally extends to three dimensions, though relatively little is done in the 3D case. Figure 2 illustrates a 3D packing problem – objects of various shape and size are packed into a rectangular container in order to minimize its dimensions. Another example is shown in Fig. 7.

The problem is NP-complete, and as a result solution methodologies pre-

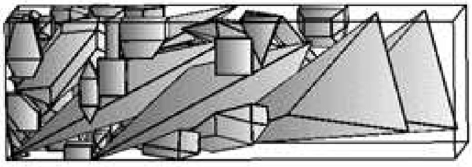

Figure 2: An example of a 3D packing problem.

dominantly utilize heuristics [5]; most existing methods of cutting and packing are restricted to objects of certain shapes and type and impose various limitations on their layout. For example, nearly all practical algorithms deal with polygons only; other shapes are simply approximated by polygons (a notable exception being [9] which allows circular shapes). Objects usually have a fixed orientation, i.e. they cannot be freely rotated. The most popular and most frequently cited tool in the modern literature on the C&P problem is the so called No-Fit Polygon (see our Section 3), it is designed to work only for polygonal objects that can be translated without rotation.

We note however that not all advances are published in academic journals because many commercial companies closely guard their products [18].

The goal of this paper is to present the results of our research group that for decades has been studying the cutting and packing problem in a formal mathematical manner. In these studies we deal with objects of very general shape (called phi-objects) and we characterize their layouts by means of special functions (called phi-functions) whose construction involves a certain degree of flexibility. The concepts of the phi-object and the phi-function have their roots in topology; but the phi-functions turn out to be highly convenient for practical solution of the C&P problem. In particular, since the construction of the phi-functions is flexible, we take advantage of this fact to develop more efficient algorithms.

While the phi-functions have been employed by our group since the 1980s, see e.g. [20], they remain little known to the broader community [5]. Our principal goal is to present here the theory of phi-objects and phi-functions in full and demonstrate practical benefits of these tools.

This paper is organized as follows: in Section 2 we introduce phi-objects, in Section 3 we define phi-functions and overview their properties, in Section 4 we use the phi-functions to reduce the C&P problem to a constrained minimization problem, and in Section 5 we describe various approaches to its solution, illustrated by example.

2 Phi-objects

Our first goal is to describe a general mathematical model for the cutting and packing (C&P) problem that should adequately represent virtually all existing applications.

The basic task is to place a set of certain geometric objects (later on, simply *objects*) T_i, $i \in \{1, 2, \ldots, n\} = I_n$, into a container T_0 so that certain restrictions on the location of the objects are met and a certain objective function (measuring the 'quality' of the placement) will reach its extreme value. We will specify these requirements below.

We can also rephrase our basic task differently: given a (large) object T_0, we need to cut a set of smaller objects $\{T_1, \ldots, T_n\}$ from it. Our objects are 2D or 3D geometric figures, i.e. subsets of $\mathbb{R}^2$ or $\mathbb{R}^3$. Generalization to any dimension is straightforward, but we do not pursue it here.

The multiplicity of shapes of T_i and T_0, as well as the variety of restrictions and forms of the objective function generate a wide specter of realizations of this basic problem. We develop a unified approach to all such applications with the ultimate goal of designing efficient algorithms for solving the C&P problem.

Phi-objects. First we define a class of admissible objects for our model following [20, 24, 25]; they will be called *ϕ-objects* or *phi-objects*. They must have interior ("main part") and boundary (frontier). Accordingly, we require each phi-object be the closure of its interior. (In mathematical topology, closed sets that are closures of their interior are said to be *canonically closed*; this is what our phi-objects are.) This requirement rules out such elements as isolated points, one-dimensional curves, etc., – they do not occur in realistic applications. Figure 3a shows an invalid phi-object – it has three one-dimensional 'whiskers', two isolated points, and four punctured interior points (white dots).

The smaller objects $T_1, \ldots, T_n$ always have finite size, in mathematical terms they are *bounded*. The (only) larger object T_0 may be unbounded (it is common in applications that the container is a strip or a cylindrical tube of infinite length).

In addition, our phi-objects should not have self-intersections along their frontier, as shown in Figure 3bc, because this may lead to confusion. For example, Figure 3c shows a dark domain whose two ends touch each other like pincers; this must be prohibited. The reason is also demonstrated in

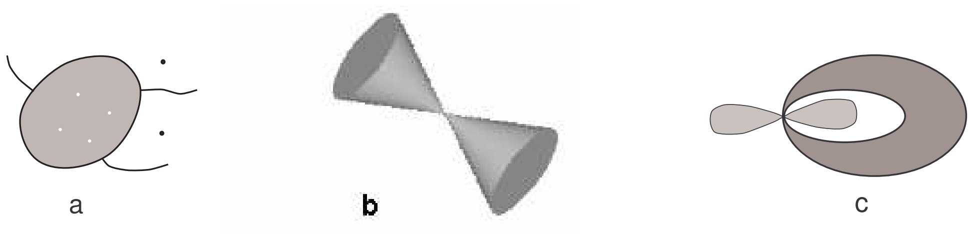

Figure 3: Invalid phi-objects.

that same figure: a similar object (the light grey "figure eight") is placed so that the two intersect each other only in their frontiers, which is generally allowed, but in this particular case we cannot position these objects as shown because one 'cuts' through the other.

Mathematically, the above requirement can be stated as follows: a phi-object and its interior must have the same homotopic type (the same number of connected components, the same number of interior holes, etc). Alternatively, one may require that for any point z on the frontier $\mathrm{fr}(T)$ of a phi-object T there exists an open neighborhood U_z of z such that $U_z \cap (\operatorname{int} T)$ is a connected set. These requirements may sound too abstract, but their practical meaning should be clear from the above example.

An important property of phi-objects is that if A is a phi-object, then the closure of its complement, i.e. $A^* = \mathrm{cl}(\mathbb{R}^d \setminus A)$, where $d = 2, 3$, is a phi-object, too.

In most applications, the frontiers of 2D phi-objects are made by simple contours: straight lines and circular arcs. Likewise, the frontiers of 3D phi-objects mostly consist of flat sides, spherical, cylindrical, and conical surfaces.

Primary and composed phi-objects. Any phi-object in $\mathbb{R}^2$ is called a *phi-polygon* if its frontier is shaped by straight lines, rays, and line segments. An ordinary polygon is a phi-polygon, but there are also unbounded phi-polygons – half-plane, a sector bounded by two intersecting lines, etc. (see illustrations in [6]).

A phi-object in $\mathbb{R}^3$ is called a *phi-polytope* if its frontier is shaped by phi-polygons. Other objects can be approximated by polygons and polytops, which is a common practice [5, 18], but we handle some curvilinear objects directly.

We call *primary* phi-object in $\mathbb{R}^2$ a *circle*, *rectangle*, *regular polygon*, or *convex polygon*. In 3D, a primary object is a *sphere*, *parallelepiped*, *circular* *cylinder*, *circular cone*, or *convex polytope*. In addition, if A is a 2D or 3D primary object, then the closure of its complement (in $\mathbb{R}^2$ or $\mathbb{R}^3$, respectively), denoted by A^*, is called a primary object, too. See some illustrations in [6].

We note that convex polygons formally include rectangles and regular polygons, but in practice it is convenient to treat the latter separately, as they can be handled more efficiently (e.g., compare (9) and (15) below).

More complex objects can be constructed from primary objects. We say that a phi-object T is *composed* if it is obtained by forming unions and intersections of primary objects, i.e.

$$T = T_1 \circ_1 T_2 \circ_2 \cdots \circ_{k-1} T_k, \tag{1}$$

where T_i are primary objects, each $\circ_i$ denotes either a union ($\cup$) or an intersection ($\cap$), and the order in which these operations are executed can be specified by a set of parentheses, for example $T = T_1 \cup (T_2 \cap T_3)$. Composed phi-objects may be very complex, see an example in Figure 4.

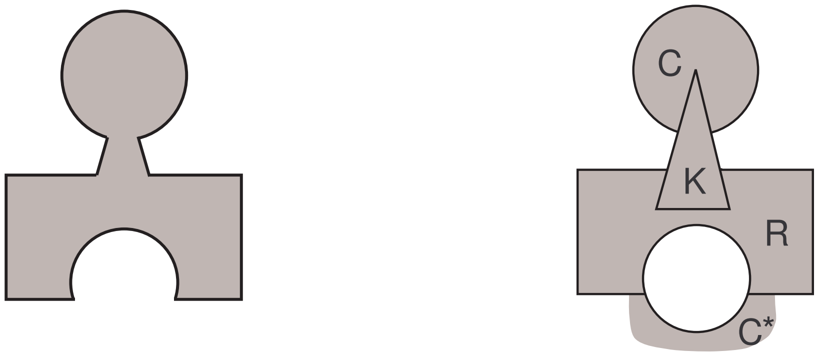

Figure 4: An example of a composed phi-object, $C \cup K \cup (R \cap C^*)$.

Fact 1. *In 2D, composed phi-objects are exactly those whose frontier is formed by straight lines, rays, line segments, and circular arcs.*

Indeed, every phi-object with such a frontier can be represented by unions and/or intersections of primary objects, in the sense of our formula (1). We will publish a formal proof of this fact separately. Similarly, 3D composed phi-objects have frontier made by flat (planar) faces, parts of spheres, and parts of cylindrical and conical surfaces, see Figure 5.

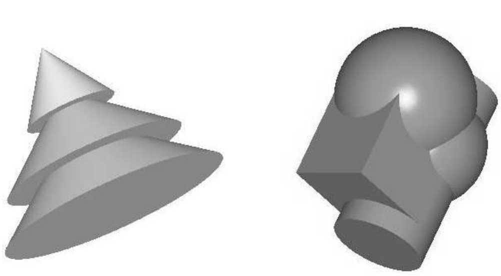

Figure 5: Examples of composed phi-objects in 3D.

Geometric parameters of phi-objects. The shape of a phi-object can be specified in many ways. For a simple (primary) object, we name its type and list its metric dimensions. For example, a circle can be specified by a pair (C, r), where C is the type ("circle") and $r > 0$ is its radius, i.e.

$$(C, r) = \{(x, y) \colon x^2 + y^2 \leq r^2\}.$$

A sphere can be specified by a pair $(\mathbb{S}, r)$, where $\mathbb{S}$ is the type ("sphere") and $r > 0$ is its radius, i.e.

$$(\mathbb{S}, r) = \{(x, y, z) \colon x^2 + y^2 + z^2 \leq r^2\}.$$

For a rectangle, we use a triple (R, a, b), where R is the type ("rectangle") and $a, b > 0$ are half-lengths of its sides:

$$(R, a, b) = \{(x, y) \colon |x| \leq a \quad \text{and} \quad |y| \leq b\}. \tag{2}$$

Similarly we describe a rectangular parallelepiped (a 3D box):

$$(\mathbb{P}, a, b, c) = \{(x, y, c) \colon |x| \leq a, \quad |y| \leq b, \quad |z| \leq c\}.$$

For a cylinder, we use a triple $(\mathbb{C}, r, h)$, where $\mathbb{C}$ is the type, and r is the radius of the base and h is the half-height:

$$(\mathbb{C}, r, h) = \{(x, y, z) \colon x^2 + y^2 \leq r^2 \quad \text{and} \quad |z| \leq h\}. \tag{3}$$

For a regular polygon, we can write (H, m, h), where H stands for the type, m denotes the number of sides and $h > 0$ is the side length. For a convex m-gon, we denote its type by K and specify its shape by a set of inequalities

$\alpha_i x + \beta_i y \leq \gamma_i$ for $i = 1, \ldots, m$; it is convenient to assume that $(0, 0)$ belongs to the polygon, then $\gamma_i \geq 0$. It is also convenient to choose α_i and β_i so that $\alpha_i^2 + \beta_i^2 = 1$, this simplifies subsequent computations. Thus the convex m-gon is described by

$$(K, \alpha_1, \beta_1, \gamma_1 \ldots, \alpha_m, \beta_m, \gamma_m). \tag{4}$$

The (closure of the) complement of a primary object is specified similarly, except we add a star to its type; for example

$$(C^*, r) = \{(x, y) \colon x^2 + y^2 \geq r^2\}.$$

Recall that this is a primary object, too.

It is important that each primary object is specified by a set of *linear* or *quadratic* inequalities. Actually quadratic formulas allow us to describe even more general shapes, such as ellipses (ovals), ellipsoids (footballs), hyperboloids (saddles), etc.

To represent a composed object, we can specify the primary objects used in its construction, their positions (in the way explained below), and the sequence of set-theoretic operations (unions and/or intersections) employed to produce the composed object from its primary constituents. The list of characteristics of a composed object may be quite long depending on the complexity of its shape.

Position parameters of phi-objects. One may notice that in our formulas that specify primary objects the origin $(0, 0)$ plays a special role. We call it a *pole* of the primary object. If the object is centrally-symmetric, then its center becomes the pole. Otherwise the choice of a pole may be quite arbitrary, for example in a generic polygon the pole can be placed at a vertex.

In addition, the orientation of the phi-object is usually fixed by its description, for example the sides of a rectangle are aligned with the coordinate axes, see (2). Thus with each phi-object we associate not only a pole but also a coordinate frame originating at the pole. We call it the *eigen* (own) coordinate system of the object. The inequalities specifying a primary object are written in their eigen coordinates.

Next in order to specify an arbitrary position of a 2D phi-object in $\mathbb{R}^2$, we introduce a translation vector $\nu = (\nu_1, \nu_2)$ and a rotation angle $\theta \in [0, 2\pi)$. This means that the object is translated by ν, i.e. its pole moves to the point (ν_1, ν_2), and then the object is rotated about the pole by θ (say, counterclockwise).

The rotation parameter θ is optional. First of all, it is redundant for such objects as circles. Second, in many applications the objects cannot be rotated freely by their nature. In garment industry, which remains the largest field of applications of cutting and packing algorithms, free rotations are generally not allowed. One cuts pieces of predetermined shape from a long strip of fabric, and there are usually just two orientations in which the pieces can be placed: the original one and the one obtained by a 180° rotation (such a restriction is due to the existence of drawing patterns and to intrinsic characteristics of the fabric's weave). We will analyze the cutting and packing problem both with and without rotation parameters.

The position of a 3D phi-object in space $\mathbb{R}^3$ requires a translation vector $\nu = (\nu_1, \nu_2, \nu_3)$ and three (optional) rotation angles $\theta_1, \theta_2, \theta_3$.

To summarize, a composed phi-object on a plane or in space can be described by a list of characteristics that include (i) types of primary objects used in its construction and the rules of construction (the sequence of intersections and/or unions), (ii) the metric dimensions of the constituent primary objects, and (iii) the translation vector and (optionally) rotation angle(s) that determine the position of the object in plane/space. While the characteristics (i) and (ii) are fixed for every phi-object, those in (iii) are usually treated as variables by the optimization algorithms which try to arrange the objects in an optimal way.

Interaction of phi-objects. In solving the cutting and packing problem it is most important to distinguish between different types of mutual location of two phi-objects (let us call them A and B):

- *Interior-intersection*: $\mathrm{int}(A) \cap \mathrm{int}(B) \neq \emptyset$.
- *Touching*: $\mathrm{int}(A) \cap \mathrm{int}(B) = \emptyset$ and $\mathrm{fr}(A) \cap \mathrm{fr}(B) \neq \emptyset$.
- *Non-intersection*: $A \cap B = \emptyset$.
- *Containment*: $A \subset B$, i.e. $\mathrm{int}(A) \cap \mathrm{int}(B^*) = \emptyset$.

We remind the reader that B^* denotes the (closure of the) complement of B. Note that the containment is conveniently described by non-intersection of the interiors of A and B^*.

3 Phi-functions

In order to formalize the above relations between phi-objects we introduce Φ-functions, or phi-functions, following [20, 24, 25]. The basic idea is that for any pair of objects A and B the phi-function $\Phi(A, B)$ must be positive for nonintersecting objects, zero for touching objects, and negative for objects with intersecting interiors. That is, $\Phi(A, B)$ must satisfy

$$(5) \qquad \begin{cases} \Phi(A, B) > 0 & \text{if} \quad A \cap B = \emptyset \\ \Phi(A, B) = 0 & \text{if} \quad \mathrm{int}(A) \cap \mathrm{int}(B) = \emptyset \quad \text{and} \quad \mathrm{fr}(A) \cap \mathrm{fr}(B) \neq \emptyset \\ \Phi(A, B) < 0 & \text{if} \quad \mathrm{int}(A) \cap \mathrm{int}(B) \neq \emptyset \end{cases}$$

Thus knowing the sign of Φ for every pair of objects would allow us to distinguish between the basic types of their mutual location. The containment $T_i \subset T_0$, in particular, holds if and only if $\Phi(T_i, T_0^*) \geq 0$.

Since the only variable characteristics of our objects are their translation vectors and (sometimes) rotation angles, the phi-function for the objects T_i and T_j can be written as $\Phi(u_i, u_j)$, where u_i and u_j denote the lists of all variable characteristics (variables) for the objects T_i and T_j, respectively. We require the phi-function be *defined* and *continuous* for all values of its variables u_i, u_j.

It is now clear that if $\Phi > 0$, then the objects are a certain distance apart; usually, decreasing Φ would bring them closer together. On the other hand, if $\Phi < 0$, then the objects overlap, and increasing Φ would force them separate. These features make the phi-functions instrumental for the performance of cutting and packing algorithms.

We remark that in some applications the metric dimensions of some objects are also variable, then they can be included in the list of arguments of the phi-functions.

Construction of phi-function. While the sign of the phi-function plays a crucial role, its absolute value may is not subject to any rigid requirements. In particular, if two objects A and B overlap, then $\Phi(A, B) < 0$, and the value $|\Phi(A, B)|$ should just roughly measure the degree of overlap. For non-overlapping objects A, B, we have $\Phi(A, B) > 0$, and the value of $\Phi(A, B)$ may just roughly correspond to the distance between A and B.

In particular, one might set $\Phi(A, B) = \mathrm{dist}(A, B)$, where

$$(6) \qquad \mathrm{dist}(A, B) = \min_{X \in A,\ Y \in B} \mathrm{dist}(X, Y)$$

denotes the geometric (Euclidean) distance between closed sets.

There is an issue of existence of the minimum (6). We recall that only one of our objects, T_0, may be unbounded itself and have an unbounded complement; thus for every pair of our objects at least one is either bounded itself or has a bounded complement; this property guarantees the existence of the above minimum.

In some cases the geometric distance between objects is easy enough to compute and it can be used as the value of the phi-function. But in many cases the formula for the distance involves radicals, which would make it difficult to use Φ and its derivatives by our local optimization algorithms. In those cases Φ should be defined by a simpler formula, which only roughly estimates the distance between the objects. We give examples below.

Phi-function for two circles. The distance between two circles C_i, $i = 1, 2$, with centers (x_i, y_i) and radii $r_i > 0$ involves a radical:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} - (r_1 + r_2).$$

We can define the phi-function is a simpler way:

$$\Phi(C_1, C_2) = (x_1 - x_2)^2 + (y_1 - y_2)^2 - (r_1 + r_2)^2. \tag{7}$$

Note that the sign of Φ coincides with that of d (and $\Phi = 0$ whenever $d = 0$). The formula (7) allows us to avoid square roots and use only quadratic functions.

Phi-function for two spheres. Similarly, for two spheres $\mathbb{S}_i$, $i = 1, 2$, with centers (x_i, y_i, z_i) and radii $r_i > 0$ we set

$$\Phi(\mathbb{S}_1, \mathbb{S}_2) = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 - (r_1 + r_2)^2. \tag{8}$$

Phi-function for two rectangles. For two rectangles R_i, $i = 1, 2$, with centers (x_i, y_i) and half-lengths of their sides $a_i, b_i > 0$ (assuming that the sides are aligned with the coordinate axes) we define the phi-function by

$$\Phi(R_1, R_2) = \max\{(|x_1 - x_2| - a_1 - a_2), (|y_1 - y_2| - b_1 - b_2)\}. \tag{9}$$

We remark that in numerical implementation of this and other formulas, the absolute value function is not used, as it makes the application of the gradient method difficult. Instead, we use minimum or maximum; for example, we compute $|x_1 - x_2| = \max\{x_1 - x_2,\ x_2 - x_1\}$.

Observe that the above function (9) sometimes coincides with the geometric distance between the rectangles (if one is above the other or they are placed side by side), but in general the distance involves square roots, while our formula is just a combination of linear functions.

Phi-function for two boxes. Similarly, for two rectangular parallelepipeds $\mathbb{P}_i$, $i = 1, 2$, with centers (x_i, y_i, z_i) and half-lengths of their sides $a_i, b_i, c_i > 0$ we set

$$\begin{aligned}\Phi(\mathbb{P}_1, \mathbb{P}_2) = \max\{&(|x_1 - x_2| - a_1 - a_2),\\ &(|y_1 - y_2| - b_1 - b_2), (|z_1 - z_2| - c_1 - c_2)\}.\end{aligned}$$

Phi-function for two cylinders. Now let $\mathbb{C}_i$, $i = 1, 2$, be two cylinders with centers (x_i, y_i, z_i), radii of the bases r_i and half-heights h_i, see (3). Combining (7) and (9) gives

$$\begin{aligned}\Phi(\mathbb{C}_1, \mathbb{C}_2) = \max\{&(|z_1 - z_2| - h_1 - h_2),\\ &(x_1 - x_2)^2 + (y_1 - y_2)^2 - (r_1 + r_2)^2\}.\end{aligned} \tag{10}$$

Phi-function for polygons. Effectively, in (9) we replace the distance between two vertices of our rectangles with the distance from a vertex of one rectangle to a side of the other (more precisely, to the line containing that side); and the distance from a point to a line is always given by a linear formula. This principle can be applied to any pair of phi-polygons.

Phi-function for convex polygons. We write Φ explicitly for convex polygons (recall that those are primary phi-objects). Suppose

$$(K', \alpha'_1, \beta'_1, \gamma'_1 \ldots, \alpha'_{m'}, \beta'_{m'}, \gamma'_{m'}). \tag{11}$$

and

$$(K'', \alpha''_1, \beta''_1, \gamma''_1 \ldots, \alpha''_{m''}, \beta''_{m''}, \gamma''_{m''}). \tag{12}$$

are two convex polygons specified according to our formula (4). Denote also by (x'_i, y'_i), $1 \le i \le m'$, the vertices of K' and by (x''_i, y''_i), $1 \le i \le m''$, the vertices of K''. As before, we assume that α_i's and β_i's satisfy $\alpha_i^2 + \beta_i^2 = 1$ for each polygon. Then the value $d = \alpha_i x + \beta_i y + \gamma_i$ is the 'signed' distance from the point (x, y) to the ith edge of the polygon; the sign of d is automatically determined as follows: it is negative if the point (x, y) lies on the same side of the edge as the entire polygon and positive otherwise.

Now let

$$u_{ij} = \alpha'_i x''_j + \beta'_i y''_j + \gamma'_i \tag{13}$$

denote the 'signed' distance from the jth vertex (x''_j, y''_j) of the polygon K'' to the ith edge of K' and

$$v_{ji} = \alpha''_i x'_j + \beta''_i y'_j + \gamma''_i \tag{14}$$

the 'signed' distance from the ith vertex (x'_i, y'_i) of the polygon K' to the jth edge of K''. We now set

$$\Phi(K', K'') = \max\Big\{ \max_{1\leq i\leq m'} \min_{1\leq j\leq m''} u_{ij}, \max_{1\leq j\leq m''} \min_{1\leq i\leq m'} v_{ji}, \Big\}. \tag{15}$$

This formula is based on two facts. The first one is a well known geometric property of convex polygons: if two convex polygons are disjoint, then there is an edge E of one of them such that these polygons lie on the opposite sides of the line containing E. This property guarantees the basic features (5) of the function (15), in particular $\Phi(K', K'') > 0$ whenever the polygons K', K'' are disjoint.

The second fact is a simple property of continuous functions: if f and g are continuous, then $\min\{f, g\}$ and $\max\{f, g\}$ are also continuous functions. This fact implies the continuity of Φ in (15).

We note that the restriction $\alpha_i^2 + \beta_i^2 = 1$ is no longer necessary as our phi-function need not represent actual distances.

If the polygons K' and K'' have fixed orientation, then their positions are completely specified by the coordinates of their poles, let us denote those by (x', y') and (x'', y''), respectively. These are the only variables in our formulas. It is easy to check that $\alpha' s_i$'s and β_i's are constants (independent of the coordinates of the poles), and γ_i's, x_i's, y_i's are just linear functions of the coordinates of the poles. Therefore the phi-function (15) is piecewise linear in its arguments (x', y') and (x'', y'').

Phi-function for polytops. In 3D space, we can apply a similar principle to phi-polytopes: replace the distance between their vertices by the distance from a vertex of one polytope to a side of the other (more precisely, to the plane containing that side); of course the vertex and the side must be properly chosen, which requires an elaborate but essentially elementary analysis. The distance from a point to a plane is always given by a linear formula.

Things may get more complicated when the frontiers of the objects are a mixture of arcs and line segments; then the constructions of phi-functions may require a degree of ingenuity, see next.

Phi-function for a rectangle and a circle. Let R be a rectangle with center (x_1, y_1) and half-lengths of its sides $a, b > 0$, and C be a circle with center (x_2, y_2) and radius $r > 0$. Then we define the phi-function by

$$(16) \qquad \Phi(R, C) = \max\big\{(u - r),\ (v - r),\ \min\{u^2 + v^2 - r^2,\ u + v - r\}\big\},$$

where $u = |x_1 - x_2| - a$ and $v = |y_1 - y_2| - b$. The reader may check by direct inspection that this Φ is continuous in x_1, y_1, x_2, y_2 and satisfies (5). Note that the phi-function is quadratic in its arguments (x_1, y_1) and (x_2, y_2).

On the other hand, the construction of phi-functions for some composed objects may turn out rather simple.

Phi-function for a 'pill' and a circle. Let P be a 'pill' (or a 'stadium'), i.e. the union of a rectangle and two circles: $P = R \cup C_1 \cup C_2$, where $R = \{|x| \leq a, |y| \leq b\}$, $C_1 = \{(x - a)^2 + y^2 = b^2\}$, and $C_2 = \{(x + a)^2 + y^2 = b^2\}$, see Figure 6; for simplicity we place the center of the pill at the origin. The other object is a circle C with center (x, y) and radius $r > 0$. Now we can define the phi-function by

$$\Phi_{P,C}(x, y) = \min\{\Phi_{R,C}, \Phi_{C_1,C}, \Phi_{C_2,C}\},$$

where $\Phi_{R,C}, \Phi_{C_1,C}, \Phi_{C_2,C}$ are defined as above.

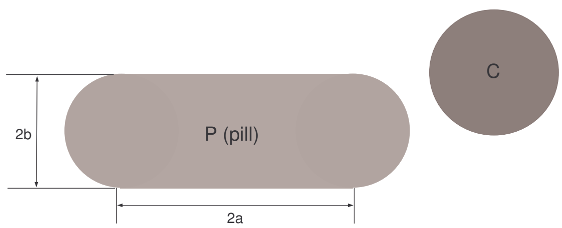

Figure 6: A 'pill' P and a circle C.

Phi-function for unions. The last example illustrates a general principle. Suppose $A = A_1 \cup \cdots \cup A_p$ and $B = B_1 \cup \cdots \cup B_q$ are composed objects, each of which is a union of some primary (or composed) objects A_i and B_j, respectively. These do not have to be disjoint unions, i.e. some A_i's may overlap, and so may some of B_j's. Then we can put

$$\Phi(A, B) = \min_{1 \leq i \leq p} \min_{1 \leq j \leq q} \Phi(A_i, B_j).$$

This fact can be verified by direct inspection, see also [6].

Phi-function for more general objects. While the construction of the phi-functions may be elaborate, it only needs to be done once for every pair of objects. In any cutting and packing problem with known shapes of available objects, one can prepare a set of properly defined phi-functions for the use by optimization algorithms. The phi-functions can be stored in advance, 'off-line', in a library, and then each instance of the problem can be solved fast by calling the ready-to-use phi-functions from the library.

It is interesting to describe pairs of phi-objects for which one can find a radical-free phi-function expressed only by linear and quadratic formulas.

Fact 2. *If A and B are 2D composed objects (i.e. their frontiers are made by straight lines, rays, line segments, and circular arcs; recall Fact 1) and we fix their orientations (i.e., exclude rotation angles), then there exists a radical-free phi-function $\Phi(A, B)$ whose formula only involves linear and quadratic expressions.*

This in fact is a mathematical theorem, and we will publish its proof separately. If the orientations of the composed objects A and B are not fixed, then the formula for $\Phi(A, B)$ will also contain sines and cosines of the rotation angles; it is still radical-free.

Normalized phi-function. Some applications involve explicit restrictions on the distances between certain pairs of objects (or between the objects and the walls of the container), i.e. some upper and/or lower limits on those distances may be set. In such cases one may need to compute exact distances between the phi-objects to meet those requirements.

Thus there may be a use for phi-functions $\Phi(A, B)$ whose values equal $\text{dist}(A, B)$ in case $A \cap B = \emptyset$. We call them *normalized* phi-functions. The computation of geometric distances between primary and composed objects may involve rather complicated formulas with radicals, see a variety of examples detailed in [6], but they all can be done by using elementary geometry, so we do not elaborate on that here.

It is also possible to avoid radicals even in this case, provided the restrictions on the distances between objects are known in advance, see the next section.

Properties of phi-functions. Suppose our objects T_1, T_2 have fixed metric characteristics and no rotation angles. Then the phi-function $\Phi(\nu_1, \nu_2)$ only depends on the two translation vectors ν_1 and ν_2. As Φ is determined by the relative position of two objects, we have

$$\Phi(\nu_1, \nu_2) = \Phi(\nu_1 - \nu_2, 0) = \Phi(0, \nu_2 - \nu_1).$$

Thus to describe the phi-function it is enough to fix the position of one object and only translate the other. Then the zero level of the phi-function, i.e.

$$\gamma_{12} = \{\nu \in \mathbb{R}^d \colon \Phi(0, \nu) = 0\}$$

(here $d = 2, 3$) plays a special role; it describes all the translations of T_2 so that it touches T_1. This set is congruent ($\simeq$) to the frontier of the Minkowski sum of the two objects, i.e. $\gamma_{12} \simeq \operatorname{fr} T_{12}$ where $T_{12} = T_1 \oplus (-1)T_2$ is the Minkowski sum. The Minkowski sum of two sets A and B is defined by

$$A \oplus B = \{X + Y \in \mathbb{R}^d \colon X \in A, Y \in B\}. \tag{17}$$

The set $\gamma_{12} \simeq \operatorname{fr} T_{12}$ is also called *shape envelope* [3] and *hodograph* [22]. We note that $\gamma_{21} \simeq -\gamma_{12}$.

Most modern studies of the C&P problem in 2D are restricted to polygons (other shapes are simply approximated by polygons) and their orientation is usually fixed, thus no rotation angles are allowed. In that case γ_{12} is also a polygon, it is called the *No-Fit Polygon* (NFP). It bounds the region where the pole of T_2 should not be placed to avoid the overlap of T_2 with T_1.

The No-Fit Polygon is the most common tool used in cutting and packing applications today, and it remains the main object of study in the modern literature on the subject. A number of efficient procedures have been developed for the construction of No-Fit Polygons; the first one was the orbiting algorithm (or sliding algorithm) of [15]. There are alternative algorithms, see [1, 7, 9, 11, 14, 26].

We note that the No-Fit Polygon coincides with the zero level set of our phi-function in the absence of rotation angles and when one object is fixed. Thus the No-Fit Polygon is a special case of the broader theory of our phi-functions [5].

We also note that if T_1 and T_2 are centrally symmetric, then their poles should be placed at their centers, and then the phi-function can (and should) be defined so that $\Phi(\nu, 0) = \Phi(0, \nu)$.

4 Mathematical optimization problem

In terms of phi-functions we can formulate the cutting and packing problem as a constrained optimization problem suitable for solving by general methods of mathematical programming. Here we do that.

First, for each object T_i we have a vector u_i of its variable parameters; these may include (i) the translation vector ν_i, (ii) the rotation angle(s) θ_i, and (iii) some metric dimensions if those are not fixed. Thus $u_0, u_1, \ldots, u_n$ constitute the variables in our model.

Objective function. The container T_0 is a special object. In most cases it is not necessary to translate or rotate it, thus we can set $\nu_0 = 0$ and $\theta_0 = 0$ and exclude these parameters from the list of variables. On the other hand, the metric characteristics of the container are usually treated as variables, as we precisely want to minimize some of those (for example, the length, or perimeter, or area, or volume of the container). Thus the general goal is to minimize a certain objective function

$$\min F(u_0, u_1, \ldots, u_n),$$

which may depend on some (or all) variables; though in most cases F only depends on the metric characteristics of T_0, i.e. $F = F(u_0)$.

Constraints. Next we list all relevant constraints. First, small objects T_i for $i = 1, \ldots, n$ must be placed in the container, i.e.

$$\Phi_{T_0^* T_i}(u_0, u_i) \geq 0 \qquad \text{for} \quad i = 1, \ldots, n$$

where T_0^* denotes the (closure of the) complement of T_0.

Second, the small objects should not overlap, i.e.

$$\Phi_{T_i T_j}(u_i, u_j) \geq 0 \qquad \text{for} \quad 1 \leq i < j \leq n.$$

Third, there may be restrictions on the minimal and/or maximal distance between certain objects, in that case we have additional constraints:

$$\rho_{ij}^- \leq \tilde{\Phi}_{T_i T_j}(u_i, u_j) \leq \rho_{ij}^+$$

for some $1 \leq i < j \leq n$; here ρ_{ij}^- denotes the minimal allowable distance and ρ_{ij}^+ the maximal allowable distance. In this case we may need to use the normalized phi-function $\tilde{\Phi}$ as the distances must be computed precisely. (But one can still avoid normalized phi-functions, see below.)

Fourth, there may be restrictions on the minimal and/or maximal distance from certain objects to the walls of the container, i.e.

$$\rho_{0i}^- \leq \tilde{\Phi}_{T_0^* T_i}(u_0, u_i) \leq \rho_{0i}^+$$

for some $1 \leq i \leq n$. Lastly, there might be constraints on the rotation angles in the form $\theta_{\min} \leq \theta \leq \theta_{\max}$. This completes the list of constraints.

We emphasize that (i) all our constraints are defined by inequalities, (ii) all our phi-functions (except the optional constraints involving maximum and minimum distances) are fairly simple – they are continuous piecewise smooth functions expressed by linear and/or quadratic formulas. The objective function F is usually simple, too (for example, it is just the length of the container).

Simplifying distance constraints. The distance constraints, as stated above, involve normalized phi-functions which may bring unwanted radicals to our analysis. However we can further simplify our formulas and eliminate radicals as follows. Suppose the minimal allowable distance ρ_{ij}^- for a pair of objects T_i, T_j is specified. Then we can construct an *adjusted* phi-function $\Phi^{\diamond}_{T_i,T_j}(u_i, u_j)$ such that

$$\Phi^{\diamond}_{T_i,T_j}(u_i, u_j) = 0 \quad \text{if and only if} \quad \tilde{\Phi}_{T_i T_j}(u_i, u_j) = \rho_{ij}^-$$

and such that the sign of $\Phi^{\diamond}_{T_i,T_j}(u_i, u_j)$ coincides with that of $\tilde{\Phi}_{T_i T_j}(u_i, u_j) - \rho_{ij}^-$. Since only the zero level set of the new function $\Phi^{\diamond}_{T_i,T_j}(u_i, u_j)$ is rigidly specified, we can define it by simpler formulas than those involved in the normalized phi-function $\tilde{\Phi}_{T_i T_j}(u_i, u_j)$, i.e. via linear and quadratic formulas only. Now the minimal distance constraint $\tilde{\Phi}_{T_i T_j}(u_i, u_j) \geq \rho_{ij}^-$ can be replaced with a simpler one

$$\Phi^{\diamond}_{T_i,T_j}(u_i, u_j) \geq 0.$$

In this way we can replace all minimal and maximal distance constraints with inequalities based on adjusted phi-functions and eliminate radicals altogether.

For primary and composed objects such a simplification is always possible. Indeed, suppose A and B are primary or composed objects and the constraint reads $\text{dist}(A, B) \geq \rho^-$. Let $A_{\rho^-} = A \oplus (C, \rho^-)$, where (C, ρ^-) denotes a circle of radius ρ^- centered at the origin and $\oplus$ is the Minkowski sum, cf. (17). The object A_{ρ^-} consists of points that are either in A or at distance $\leq \rho^-$ from A, and it is clearly a composed object, too.

Now the original constraint $\text{dist}(A, B) \geq \rho^-$ can be replaced by an equivalent one: $\Phi(A_{\rho^-}, B) \geq 0$, and due to the fact cited in the previous section there exists a phi-function $\Phi(A_{\rho^-}, B)$ which can be constructed without radicals.

General remarks. All our constraints define an admissible region W in the space of all the variables $u_0, u_1, \ldots, u_n$. The region W is also called the *solution space*. We make a few remarks:

1. Our constraint optimization problem is NP-hard and always has multiple local extrema.

2. The solution space W is a disconnected set. Each connected component of W may have a complicated structure, in particular it may have multiple internal holes, through holes, and cavities.

3. The solution space W can be naturally represented as $W = \cup_{j=1}^{J} W_j$, where each W_j is specified by a system of inequalities involving our phi-functions. It should be noted that J (the number of W_j's) may be huge, such as $J \sim 10^5$ or 10^{10}.

4. The frontier of W is usually made of nonlinear surfaces containing valleys, ravines, etc.

We outline various solutions of this optimization problem in the next section.

5 Solving the optimization problem

Here we discuss various approaches to the solution of the optimization problem described in the previous section, i.e. finding a global minimum (or at least a good approximation to it) of the objective function F.

We treat this task as a mathematical minimization problem. Given an initial approximation, i.e. a point $U = (u_0, u_1, \ldots, u_n)$ in the solution space W, our algorithm performs a local search, i.e. moves (modifies) the point $U \in W$ attempting to find a local minimum of the function F.

A point $U \in W$ corresponds to a particular layout of all the objects $T_1, \ldots, T_n$ within T_0, and moving the point U through W means a *simultaneous* motion of all the objects $T_1, \ldots, T_n$ in T_0. This is where our algorithm differs from many others – in most existing applications one only moves one or two objects at a time, since a simultaneous motion of all the objects is regarded as a prohibitively complicated task.

We are able to move all the objects at once, i.e. perform a local search in the multidimensional solution space W, because of our use of phi-functions. We remind the reader that our phi-functions are continuous and piecewise-smooth, and in most practical cases they are conveniently defined by simple (linear and quadratic) formulas. These features are essential for smooth performance of local minimization schemes. Our local search employs a modification of the Zoutendijk algorithm of feasible directions [27].

Thus, given an initial point $U_1 = (u_0, u_1, \ldots, u_n) \in W$, our algorithm finds a local minimum of the objective function. More precisely, given $U_1 \in W$ it forms a natural subset $W_{j,1} \subset W$ of the solution space W containing U_1, i.e. $U_1 \in W_{j,1}$. Then it finds $U_2 \in W_{j,1}$ such that $F(U_2) \leq F(U)$ for all $U \in W_{j,1}$ (or at least in a vicinity of U_2), and in addition $F(U_2) < F(U_1)$. Then, given U_2, the algorithm forms another natural subset $W_{j,2} \subset W$ containing U_2, i.e. $U_2 \in W_{j,2}$, and finds a point U_3 such that $F(U_3) \leq F(U)$ for all $U \in W_{j,2}$ (or at least in a vicinity of U_3) and $F(U_3) < F(U_2)$. We repeat this procedure until a local minimum of the objective function is found.

To find a global minimum of F in the whole space W one would need an exhaustive search, i.e. a search over *every* subset $W_j \subset W$, which is an unrealistic task, because the number of those components may be of order 10^5 or 10^{10}. In practice, only a few (well chosen) initial points $U_1, \ldots, U_k \in W$ may be examined, so the task of choosing *good initial layouts* becomes of paramount importance.

In many industrial applications, experienced workers “manually” (with the help of CAD systems) build a high quality layout, see e.g. [13], which can be then followed by a quick run of a computer optimization program to improve the manual layout as much as (locally) possible.

However in many other applications there are no “expert layouts” available, and one has to rely on computer generated initial arrangements. To this end various heuristics (and ‘metaheuristics’) are used, the simplest and most popular perhaps being the *bottom-left placement procedure*. It places objects, one by one, in the most bottom-left vacant corner of the container. When positioning an object, the procedure takes into account the previously placed objects, first to avoid overlaps, and then (in some implementations) to fill holes left empty at earlier stages. Gomes and Oliveira [12] also propose a randomized version of this method, where at each step the object to be placed next is selected randomly with probability proportional to its length.

Many authors then use various heuristics to (globally) alter the initial layout to obtain other layouts (and thus reach different components of the solution space W). One can swap two randomly chosen objects, or apply more sophisticated strategies such as 'tabu search algorithms' [2, 4] or simulated annealing [13, 19], or various genetic algorithms [10].

In some implementations, objects are allowed (temporarily) to overlap and move through one another, so that the algorithm can perform a wider search over the solution space W. In that case one needs to estimate (and penalize) the degree of overlap of objects so that the algorithm will gradually move them apart (separate) and arrive at an admissible layout (with no overlaps) in the end. In this respect our phi-functions may be useful, too, as they provide such a feature as an estimate of the degree of overlap. Other authors develop different tools to penalize overlap; see [4, 13, 14].

We generate good initial layouts as follows, see [21]. First, we approximate the container T_0 and objects $T_1, \ldots, T_n$ by rectangular polygons $P_0, P_1, \ldots, P_n$ with sides parallel to two fixed coordinate axes. Then we place the polygonal figures $P_1, \ldots, P_n$ into P_0 consecutively, according to an object sequence $P_{i_1}, \ldots, P_{i_n}$ generated by a modification of the decremental neighborhood method. This procedure employs a probabilistic search and is designed to find the most promising ones. The latter will correspond to some points $U_1, \ldots, U_k$ in W, and their number must be kept small, say $k \sim 5$ or 10. The time consuming search for local minima of F is only applied to the best initial points $U_1, \ldots, U_k$, and it produces k local minima $U_1^*, \ldots, U_k^*$ of F. In the end, we choose the local minimum of F where the value of F is smaller than at the other local minima, i.e. we choose $U^* = U_m^*$, where $m = \operatorname{argmin}\{F(U_i^*), 1 \leq i \leq k\}$. This completes our algorithm.

6 Numerical examples

We illustrate our techniques by two model examples, which have been obtained recently and were not reported yet.

First is the problem of packing cylinders of various shapes and sizes into a rectangular box, see Figure 7. We assume that the number of cylinders and their metric characteristics are given, i.e. we have $T_i = (\mathbb{C}_i, r_i, h_i)$, $i = 1, \ldots, n$, according to (3). The container is a parallelepiped

$$\mathbb{P} = \{(x, y, z)\colon |x| \leq a, \quad |y| \leq b, \quad |z| \leq c\}$$

of a fixed base $2a \times 2b$ but variable height $2c > 0$. The goal is to pack all the cylinders into $T_0 = (\mathbb{P}, a, b, c)$ in order to minimize its height $2c$.

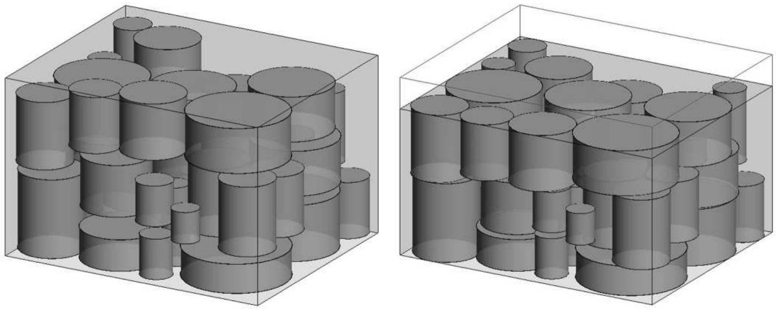

Figure 7: Packing cylinders into a box: a randomly generated initial placement (left) and the computed locally optimal arrangement (right).

In example shown in Figure 7, we have $n = 40$ cylinders. They are vertically oriented, so we do not have to rotate them. The parameters in this case are translation vectors $\nu_i = (x_i, y_i, z_i)$, $1 \le i \le n$, that specify the position of the cylinders in $\mathbb{P}$. Since c is a variable, too, we have a total of $3n+1$ variables in the model, and our solution space W is a subset of $\mathbb{R}^{3n+1}$.

The objective function to be minimized is

$$F(c, x_1, y_1, z_1, \ldots, x_n, y_n, z_n) = c.$$

We have two types of constraints. First,

$$\Phi_{\mathbb{P}^*\mathbb{C}_i}(c, u_i) \ge 0 \qquad \text{for} \ \ i = 1, \ldots, n \tag{18}$$

where $\mathbb{P}^*$ denotes the (closure of the) complement of $\mathbb{P}$. This constraint means that the cylinder $\mathbb{C}_i$ lies wholly inside $\mathbb{P}$, but may touch the walls of $\mathbb{P}$. Second,

$$\Phi_{\mathbb{C}_i\mathbb{C}_j}(u_i, u_j) \ge 0 \qquad \text{for} \ \ 1 \le i < j \le n. \tag{19}$$

which means that the cylinders $\mathbb{C}_i$ and $\mathbb{C}_j$ do not overlap (but are allowed to touch each other).

The phi-function in (18) can be computed as

$$\Phi_{\mathbb{P}^*\mathbb{C}_i}(c, u_i) = \min\{(a - |x_i| - r_i),\ (b - |y_i| - r_i),\ (c - |z_i| - h)\}.$$

The phi-function in (19) is computed according to our early formula (10).

Figure 7 shows the output of our algorithm: an initial placement generated randomly (on the left) has height $c_{\text{ini}} = 7.913$, and the computed arrangement (on the right) has height $c_{\min} = 6.5$. A similar problem is discussed in [23], but they use the normalized phi-function that involves radicals.

In the second example, we place $n = 32$ irregular 2D objects into a rectangular strip of a fixed height and variable length:

$$S = \{(x, y) \colon 0 \le y \le 12, \quad 0 \le x \le L\},$$

where L is a variable to be minimized.

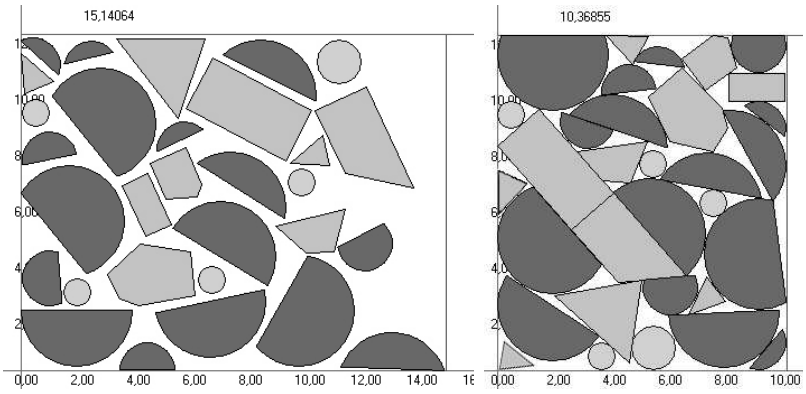

Figure 8: Packing irregular 2D objects into a strip: a randomly generated initial placement (left) and the computed optimal arrangement (right).

The objects include 11 convex polygons (triangles, quadrilaterals, a pentagon, and a hexagon) of various shapes, 16 circular segments and 5 full circles (disks) of various sizes, see Figure 8. The objects are allowed to move within the strip and rotate freely. The variables in the model include 32 translation vectors $\nu_i = (x_i, y_i)$, $1 \le i \le 32$, and 27 rotation angles θ_i, $1 \le i \le 27$ (we only need rotation angles for polygons and circular segments, as 5 disks need no rotation), plus one parameter, L, for the strip. The total number of variables in the model is $64 + 27 + 1 = 92$.

The objective function to be minimized is

$$F(L, x_1, y_1, \theta_1, \ldots, x_{32}, y_{32}) = L.$$

We have two types of constraints: first, the the objects must lie wholly inside the strip S but may touch the walls of S, and second, the objects must not overlap but are allowed to touch each other. These constraints are given in terms of the respective phi-functions, some of them were discussed in Section 3. We omit explicit formulas for the sake of brevity.

The initial placement is shown in Figure 8 (left); it is generated randomly and is confined in a rectangle of length $L_{\text{initial}} = 15.14064$. The arrangement corresponding to a local minimum of F, as computed by our algorithm, is shown in Figure 8 (right); all the 32 objects are tightly packed into a rectangle of length $L_{\min} = 10.36855$. The size of the rectangle is reduced by about 50%. Observe that the algorithm rearranged the objects and rotated many of them – the final packing has little resemblance of the initial placement.

The above examples demonstrate the potential of our algorithm to work with a large number of irregular objects in 2D and 3D and achieve tight packing arrangements that are hard to find otherwise, especially by manual work.

7 Conclusions

We demonstrate how the use of phi-functions and mathematical programming can improve the performance of cutting and packing algorithms. Our phi-functions have the following features:

- They can be applied to 2D and 3D objects of very general type (phi-objects); these include disconnected objects, non-convex objects, regions with holes and cavities, etc.
- Our phi-functions take into account continuous translations and rotations of objects.
- They may take into account variable metric characteristics of objects.
- They take into account possible restrictions on the (minimal and/or maximal) distances between objects and from the objects to the walls of the container.
- Our phi-functions are useful when dealing with overlapping objects, as they measure the degree of overlap.
- In most practical cases, the phi-functions (unlike geometric distances) are defined by simple (linear and quadratic) formulas, which allows us to use optimization algorithms of mathematical programming.

- Overall, our phi-functions allow us to enlarge the class of optimization placement problems that can be effectively solved.

Acknowledgement. The authors would like thank Drs A. Chugay and M. Zlotnik, as well as graduate student E. Stupak, for preparing some examples in this paper. N.C. was partially supported by National Science Foundation, grant DMS-0652896.

References

[1] P. K. Agarwal, E. Flato, D. Halperin, Polygon Decomposition for Efficient Construction of Minkowski Sums, Comp. Geometry Theory Appl. 21 (2002) 29–61.

[2] A. Albano, G. Sapuppo, Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Methods, IEEE Transl. System Man Cybernetics, 10 (1980) 242–248.

[3] R. C. Art, An Approach to the Two-Dimensional Irregular Cutting Stock Problem, Tech. Report 36.Y08, IBM Cambridge Scientific Centre, Report, 1966.

[4] J. A. Bennell, K. A. Dowsland, Hybridising Tabu Search with Optimisation Techniques for Irregular Stock Cutting, Management Science 47 (2001) 1160–1172.

[5] J. A. Bennell and J. F. Oliveira, The geometry of nesting problems: A tutorial, Europ. J. Operational Research, 184 (2008) 397–415.

[6] J. Bennell, G. Scheithauer, Yu. Stoyan, T. Romanova, Tools of mathematical modelling of arbitrary object packing problems, J. Ann. Operations Research, 2008, in press.

[7] J. A. Bennell, X. Song, A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums, Computers and Operations Research, 35 (2008), 267–281.

[8] E. G. Birgin, J. M. Martinez, F. H. Nishihara, D. P. Ronconi, Ortogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization, Computers & Operations Research 33 (2006) 3535–3548.

[9] E. Burke, R. Hellier, G. Kendall, G. Whitwell, A New Bottom-Left-Fill Heuristic Algorithm for the Two-Dimensional Irregular Packing Problem, Operations Research, to appear.

[10] R. Dighe, M. J. Jakiela, Solving Pattern Nesting Problems with Genetic Algorithms Employing Task Decomposition and Contact, Evolutionary Computation, 3 (1996) 239–266.

[11] P. K. Ghosh, An algebra of polygons through the notion of negative shapes, CVGIP: Image Understanding, 54 (1991) 119–144.

[12] A. M. Gomes, J. F. Oliveira, A 2-exchange heuristic for nesting problems, European J. Operational Research, 141 (2002) 359–370.

[13] A. M. Gomes, J. F. Oliveira, Solving Irregular Strip Packing problems by hybridising simulated annealing and linear programming, European J. Operational Research, 171 (2006) 811–829.

[14] Z. Li, V. Milenkovic, Compaction and separation algorithms for non-convex polygons and their applications, European J. Operational Research, 84 (1995) 539–561.

[15] D. A. Mahadevan, Optimization in Computer-Aided Pattern Packing, Ph.D. Thesis, North Carolina State University, 1984.

[16] V. Milenkovic, Rotational polygon overlap minimization and compaction, Computational Geometry, 10 (1998) 305–318.

[17] V. Milenkovic, Rotational polygon containment and minimum enclosure using only robust 2D constructions, Computational Geometry, 13 (1999) 3–19.

[18] V. J. Milenkovic, K. Daniels, Translational polygon containment and minimal enclosure using mathematical programming, Intern. Trans. Operational Research, 6 (1999) 525–554.

[19] J. F. Oliveira, J. S. Ferreira, Algorithms for Nesting Problems, Applied Simulated Annealing, In: R. Vidal (Ed.), Lect. Notes Econ. Math. Syst., 396, Springer-Verlag, 1993, pp. 255–274.

[20] Yu. Stoyan, Mathematical methods for geometric design, In: Advances in CAD/CAM, Proceedings of PROLAMAT82 (Leningrad, USSR, May 1982), North-Holland, Amsterdam, 1983, pp. 67–86.

[21] Y. Stoyan, N. I. Gil, G. Scheithauer, A. Pankratov, and I. Magdalina, Packing of convex polytopes into a parallelepiped, Optimization, 54 (2005), 215–235.

[22] Y. G. Stoyan, L. D. Ponomarenko, Minkowski sum and hodograph of the dense placement vector function, Reports Ukrainian SSR Acad. Sci., Ser. A, 10 (1977).

[23] Y. G. Stoyan and A. Chugay, Packing cylinders and rectangular parallelepipeds with distances between them, European J. Oper. Research (2008), in press.

[24] Y. Stoyan, G. Scheithauer, N. Gil, T. Romanova, Φ-functions for complex 2D-objects, 4OR: Quarterly J. Belgian, French and Italian Operations Research Soc. 2 (2004) 69–84.

[25] Yu. Stoyan, J. Terno, G. Scheithauer, N. Gil, T. Romanova, Phi-functions for primary 2D-objects, Studia Informatica Universalis, 2 (2001) 1–32.

[26] P. D. Watson, A. M. Tobias, An efficient algorithm for the regular W_1 packing of polygons in the infinite plane. J. Operational Research Soc., 50 (1999) 1054–1062.

[27] G. Zoutendijk, Methods of feasible directions, A study in linear and non-linear programming. Elsevier, New York, 1960.