Research Article

Phi-Functions for 2D Objects Formed by Line Segments and Circular Arcs

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

[1] *Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL 35294, USA*
[2] *Department of Mathematical Modeling, Institute for Mechanical Engineering Problems of*
   The National Academy of Sciences of Ukraine, Kharkov, Ukraine

Correspondence should be addressed to N. Chernov, chernov@math.uab.edu

We study the cutting and packing (C&P) problems in two dimensions by using phi-functions. Our phi-functions describe the layout of given objects; they allow us to construct a mathematical model in which C&P problems become constrained optimization problems. Here we define (for the first time) a complete class of basic phi-functions which allow us to derive phi-functions for *all* 2D objects that are formed by linear segments and circular arcs. Our phi-functions support translations and rotations of objects. In order to deal with restrictions on minimal or maximal distances between objects, we also propose adjusted phi-functions. Our phi-functions are expressed by simple linear and quadratic formulas without radicals. The use of radical-free phi-functions allows us to increase efficiency of optimization algorithms. We include several model examples.

1. Introduction

We study the cutting and packing (C&P) problems. Its basic goal is to place given objects into a container in an optimal manner. For example, in garment industry one cuts figures of specified shapes from a strip of textile, and one naturally wants to minimize waste. Similar tasks arise in metal cutting, furniture making, glass industry, shoe manufacturing, and so forth. In shipping works one commonly needs to place given objects into a container of a smallest size or volume to reduce the space used or increase the number of objects transported.

The C&P problem can be formally stated as follows: place a set of given objects $A_1, \ldots, A_n$ into a container Ω so that a certain objective function (measuring the "quality" of placement) will reach its extreme value.

In some applications (as in garment industry) objects must be specifically oriented respecting the structure of the textile, that is, they can only be translated without turnings (or only slightly rotated within given limits). In other applications objects can be freely rotated. Some applications involve additional restrictions on the minimal or maximal distances between certain objects or from objects to the walls of the container Ω (one example is packing of radioactive waste).

While most researchers use heuristics for solving C&P problems, some develop systematic approaches based on mathematical modeling and general optimization procedures; see, for example, [1–3]. We refer the reader to recent tutorials [4, 5] presenting the history of the C&P problem and basic techniques for its solution.

Standard existing algorithms are restricted to 2D objects of polygonal shapes; other shapes are simply approximated by polygons (a notable exception is [6] which also treats circular objects). The most popular and most frequently cited tools in the modern literature on the C&P problem are Minkowski sum [7] and the so-called No-Fit Polygon [4], which works with polygons only and does not support rotations. Rotations of polygons were considered in [8, 9], and in a very recent paper [10] the concept of No-Fit Polygon was extended to objects bounded by circular arcs.

In this paper we develop tools that handle any 2D objects whose boundary is formed by linear segments and/or circular arcs (the latter may be convex or concave). All objects we had to deal with in real applications, without exception, belong to this category. Our tools support free translations and rotations of objects and can respect any restrictions on the distances between objects.

We describe the layout of objects relative to each other by the so-called *phi-functions*. For any placement of two objects A_i and A_j on the plane $\mathbb{R}^2$, the corresponding phi-function $\Phi^{A_i A_j}$ shows how far these objects are from each other, whether they touch each other, or whether they overlap (in the latter case it shows how large the overlap is). Phi-functions were introduced in [11–13] and fully described in our recent survey [14]. Phi-functions are also used for solving 3D packing problems [15] and covering problems [16].

The arguments of the phi-function $\Phi^{A_i A_j}$ are the translation and rotation parameters of the objects A_i and A_j; those parameters specify the exact position and orientation of the objects in the xy plane (or in the xyz space). All these parameters together, for all the given objects, constitute the *solution space*. Solving the cutting and packing problem then consists of minimization of a certain objective function defined on the solution space.

Thus the solution of the C&P problem reduces to minimization of an objective function on a certain (multidimensional) space, which can be done by mathematical programming. A detailed description of the solution strategy is given in [14]. We emphasize that the minimization is performed with respect to all of the underlying variables, that is, all the objects can move and rotate simultaneously. In this respect our approach differs from many others that optimize the position of one object at a time. Illustrations and animated demonstrations of the performance of our methods can be found on our web page [17].

The solution space consists of all admissible (nonoverlapping) positions of our objects, which correspond to inequalities $\Phi^{A_i A_j} \geq 0$ for all $i \neq j$. Our phi-functions $\Phi^{A_i A_j}$ are defined by a combination of minima and maxima of various basic functions that represent mutual position of various elements of the underlying objects (their edges, their corner points, etc.) As a result, the solution space is described by a complicated tree in which each terminal node consists of a system of inequalities involving translation and rotation parameters of certain objects. This description is very complex, and one of our goals is to simplify it.

The above-mentioned inequalities may be expressed via distances between various points, segments, and arcs on the boundaries of our objects. The resulting formulas often involve square roots, which may cause unpleasant complications—formulas describing the solution space develop singularities, and the minimization process becomes prohibitively slow.

To remedy the situation, we redefine the phi-functions so that the solution space will be described by simpler formulas without radicals (thus avoiding related singularities), This speeds up the optimization process. By our rules, phi-functions only need to satisfy certain flexible requirements, they are not rigidly determined by the shapes of the given objects. In fact, one can often define phi-functions for fairly complicated objects by simple formulas that avoid square roots and other irrational functions.

This strategy was employed in our previous works [14], but here we implement it to an utmost extent. We will show that for *any* objects bounded by linear segments and circular arcs phi-functions can be defined by algebraic formulas without radicals. This is the principal goal of our paper. It was announced in [14] without much details. Here we give explicit practical formulas for computing the phi-functions in all possible cases. Our radical-free phi-functions also incorporate additional constraints on the distances between objects (see Section 4).

We demonstrate the efficiency of our new phi-functions by model examples. For the description of the solution space via phi-functions we refer the reader to [14]. For further details of local optimization algorithms used in our programs we refer the reader to [18].

2. Phi-Functions and Decomposition of Objects

Recall that a 2D object is a subset $A \subset \mathbb{R}^2$; it is usually specified by some equations or inequalities in the canonical coordinates x, y. Placing the object in $\mathbb{R}^2$ means moving it without changing its shape or size. Rigid motions in $\mathbb{R}^2$ consist of rotations and translations. If we rotate A by angle θ_A (say, clockwise) and translate it by vector $v_A = (v_{Ax}, v_{Ay})$, then the resulting set (*placed object*) can be described by equation

$$A(v_A, \theta_A) = R_{\theta_A} A + v_A, \tag{2.1}$$

where

$$R_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{2.2}$$

denotes the standard rotation matrix. We call v_A and θ_A the *placement parameters* for the object A.

Now let $A, B \subset \mathbb{R}^2$ be two objects. We denote the corresponding placed objects by $A' = A(v_A, \theta_A)$ and $B' = B(v_B, \theta_B)$. The phi-function

$$\Phi = \Phi^{AB} = \Phi^{AB}(v_A, \theta_A, v_B, \theta_B), \tag{2.3}$$

describes the mutual position (interaction) of the pair of sets A' and B'. It must satisfy three basic requirements:

$$\begin{aligned}
&\Phi > 0 \quad \text{if } A' \cap B' = \emptyset, \\
&\Phi = 0 \quad \text{if } \text{int}(A') \cap \text{int}(B') = \emptyset \text{ and } \partial A' \cap \partial B' \neq \emptyset, \\
&\Phi < 0 \quad \text{if } \text{int}(A') \cap \text{int}(B') \neq \emptyset.
\end{aligned} \tag{2.4}$$

Here $\text{int}(A')$ denotes the interior of A' and $\partial A'$ the boundary (frontier) of A'. We always assume that our objects are canonically closed sets, that is, each object is the closure of its interior. Also, the boundary $\partial A'$ should not have self-intersections [11, 14].

Note that Φ^{AB} is a function of six real variables $v_{Ax}, v_{Ay}, \theta_A, v_{Bx}, v_{By}, \theta_B$. An important requirement is that Φ^{AB} is continuous in all these six variables [14]. We will also assume that Φ^{AB} is "symmetric" in the sense that

$$\Phi^{AB}(v_A, \theta_A, v_B, \theta_B) = \Phi^{BA}(v_B, \theta_B, v_A, \theta_A) \tag{2.5}$$

and translation invariant, that is, for any vector v

$$\Phi^{AB}(v_A, \theta_A, v_B, \theta_B) = \Phi^{AB}(v_A + v, \theta_A, v_B + v, \theta_B). \tag{2.6}$$

Our phi-functions are also rotation invariant in a natural sense. In our formulas, the superscripts of Φ will always refer to given objects, while the arguments of Φ (the placement parameters) will be often omitted for brevity.

The general meaning of (2.4) is that when the placed objects are disjoint, that is, a positive distance apart, then $\Phi > 0$. When those objects just touch each other (on their boundaries), but do not overlap, then $\Phi = 0$. When they overlap, then $\Phi < 0$.

We emphasize that the exact value of the phi-function is not subject to any rigid constraints. If two placed objects A' and B' are disjoint, then Φ should just roughly approximate the distance between them. If they overlap, then the absolute value $|\Phi|$ should just roughly measure the extent of overlap. This flexibility allows us to construct relatively simple phi-functions for rather complex objects, which is the main goal of our paper.

For example, let C_1 and C_2 be two circles (disks) of radii r_1 and r_2, respectively, defined by

$$C_i = \left\{ (x, y) : x^2 + y^2 \leq r_i^2 \right\}. \tag{2.7}$$

Now by translating C_1 and C_2 through some vectors v_1 and v_2, we get two placed circles C_1' and C_2' with centers (v_{1x}, v_{1y}) and (v_{2x}, v_{2y}), respectively, and the same radii r_1 and r_2 (rotations are redundant for circles). Now the distance between C_1' and C_2' is $d = \max\{\phi, 0\}$, where

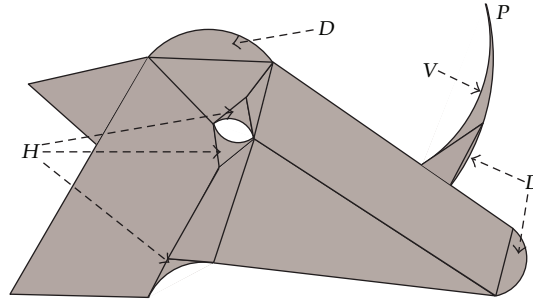$$\phi = \sqrt{(v_{1x} - v_{2x})^2 + (v_{1y} - v_{2y})^2} - (r_1 + r_2). \tag{2.8}$$

Figure 1: A partition of an object into 15 basic objects: 8 convex polygons, 3 circular segments (marked by D), 3 hats (marked by H), and one horn (marked by V).

Note that $\phi > 0$ if the circles are disjoint and $\phi < 0$ if they overlap, thus we could set $\Phi^{C_1 C_2} = \phi$. But we define $\Phi^{C_1 C_2}$ differently:

$$\Phi^{C_1 C_2} = (v_{1x} - v_{2x})^2 + (v_{1y} - v_{2y})^2 - (r_1 + r_2)^2. \tag{2.9}$$

Note that the sign of $\Phi^{C_1 C_2}$ coincides with that of ϕ (and $\Phi^{C_1 C_2} = 0$ whenever $\phi = 0$). But the formula (2.9) allows us to avoid radicals, thus improving the performance of our optimization algorithms.

Next suppose

$$A = A_1 \cup \cdots \cup A_p, \qquad B = B_1 \cup \cdots \cup B_q \tag{2.10}$$

are two objects, each of which is a union of some smaller (and simpler) components A_i and B_j, respectively. Those do not have to be disjoint, that is, some A_i's may overlap, and so may some of the B_j's. When the object A is rotated and translated, all its parts are rotated by the same angle and translated through the same vector, so the placement parameters for A and for all its parts A_i are the same. This applies to B and its parts, too.

Now we can define

$$\Phi^{AB} = \min_{1 \leq i \leq p} \min_{1 \leq j \leq q} \Phi^{A_i B_j}. \tag{2.11}$$

This simple fact can be verified by direct inspection, see also [11, 14].

In this paper we consider objects whose boundary is formed by linear segments and/or circular arcs (the latter may be convex or concave); see an example in Figure 1. Such objects can be partitioned into simpler components of four basic types: (a) convex polygons, (b) circular segments, (c) "hats", and (d) "horns"; see Figure 2.

A convex polygon is an intersection of $m \geq 3$ half-planes. More generally, an intersection of $m \geq 1$ half-planes will be called a generalized convex polygon, or phi-polygon. It may be a regular (bounded) polygon, or an unbounded region, such as a region between two rays (half-lines) emanating from a common vertex (see illustrations in [11]).

A circular segment is a region bounded by a circular arc (smaller than a semicircle) and the respective chord. One can also describe a circular segment as the convex hull of a circular
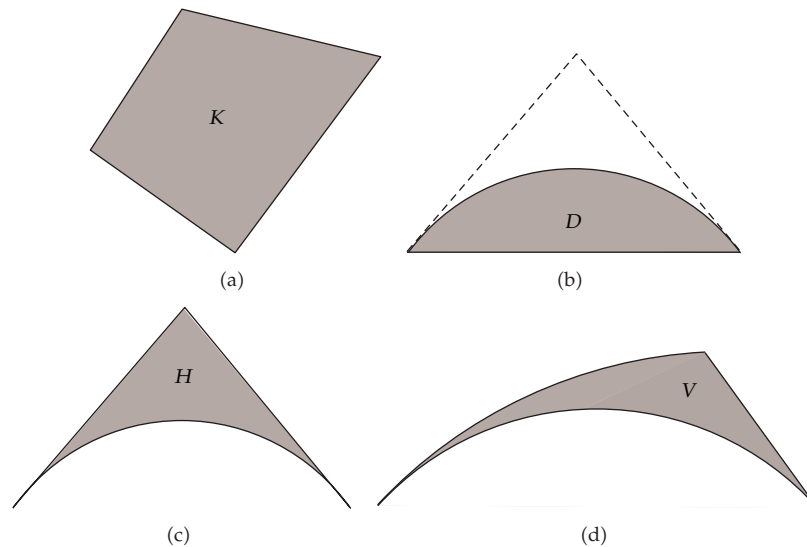
Figure 2: Basic objects: (a) convex polygon K, (b) circular segment D, (c) hat H, and (d) horn V.

arc. A *hat* is formed by a circular arc (smaller than a semicircle) and two tangent lines at its endpoints (Figure 2(c)). A *horn* is made by two circular arcs (one convex and one concave) that are tangent to each other at the point of contact and a line crossing both arcs and tangent to the concave one (Figure 2(d)). We will denote these four types by K, D, H, and V, as in Figure 2.

Figure 1 shows a division of an object into basic subobjects. It consists of 8 convex polygons, 3 circular segments, 3 hats, and one horn.

Decomposition of a given object into basic subobjects can be done by a computer algorithm based on the following steps.

(1) Locate "beaks," that is, points on the boundary of A where two arcs (one concave and one convex) terminate with a common tangent line. At each beak, cut off a small piece that is shaped as a horn (by a line tangent to the concave arc). After the detachment of horns, the resulting object will have no beaks.

(2) Locate all concave arcs and carve out hats so that each concave arc will be replaced with a polygonal line. After the detachment of hats, the resulting object will have no concave arcs.

(3) Locate all convex arcs and cut off circular segments so that each convex arc will be replaced with one or more chords. After the detachment of segments, the resulting object will have no convex arcs.

(4) If the resulting phi-polygon is convex, keep it. If not, decompose it into two or more convex ones.

We note that if the given object A is simple enough, it may not be necessary to divide it into basic objects. For example, if A is a circle (or a ring), there is no need to cut it artificially into some polygons and circular segments, as phi-functions for circles (and rings) are quite simple; see (2.9) and other formulas below, as well as [11–13].

Next, recall that our basic goal is to place given objects $A_1, \ldots, A_n$ into a container Ω with respect to a given objective. To ensure that the placed objects $A_i(v_{A_i}, \theta_{A_i})$ do not overlap, we can just verify that $\Phi^{A_i A_j} \geq 0$ for all $i \neq j$. To ensure that $A_i(v_{A_i}, \theta_{A_i}) \subset \Omega$, we verify that $\Phi^{A_i \Omega^*} \geq 0$, where $\Omega^* = \mathrm{cl}(\mathbb{R}^2 \setminus \Omega)$ is the closure of the complement to the container Ω. This is a part of our optimization algorithm; see [14].

The necessity of treating $\Omega^* = \mathrm{cl}(\mathbb{R}^2 \setminus \Omega)$ as a (rather special) object leads us to considering unbounded objects, too. Given a bounded object B, we denote by $B^* = \mathrm{cl}(\mathbb{R}^2 \setminus B)$ the unbounded complementary object. If B^* is delimited by line segments and circular arcs, then it can be decomposed into basic objects of the same four types, except one or more basic objects are unbounded phi-polygons as described above.

3. Basic Phi-Functions

Due to the decomposition principle (2.11) our problem reduces the construction of phi-functions for all pairs of basic objects. As there are four types of basic objects, there are a total of $4 + \binom{4}{2} = 10$ possible pairs of types of basic objects to treat. These will form a complete class of basic phi-functions.

3.1. Two Convex Polygons

A convex polygon is an intersection of several half-planes. A half-plane can be defined by $\alpha x + \beta y + \gamma \leq 0$ so it is completely specified by three parameters (α, β, γ). Without loss of generality we assume in what follows that $\alpha^2 + \beta^2 = 1$. A convex polygon (phi-polygon) K that is the intersection of m half-planes can be specified by

$$K = \left((\alpha_1, \beta_1, \gamma_1), \ldots, (\alpha_m, \beta_m, \gamma_m) \right). \tag{3.1}$$

Alternatively, K can be specified by a sequence of vertices

$$K = \left((x_1, y_1), \ldots, (x_m, y_m) \right) \tag{3.2}$$

listed in the counterclockwise direction. If the polygon K is moved (rotated and translated), its parameters $\alpha_i, \beta_i, \gamma_i$ and x_i, y_i can be recomputed in terms of the rotation angle θ_K and translation vector v_K, according to (2.1). Thus the placement parameters of K can be incorporated into $\alpha_i, \beta_i, \gamma_i$, and x_i, y_i.

Now let K be a convex m-gon and K' another convex m'-gon whose parameters we denote by $(\alpha_i', \beta_i', \gamma_i')$ and whose vertices are denoted by (x_i', y_i') for $1 \leq i \leq m'$. Denote

$$u_{ij} = \alpha_i x_j' + \beta_i y_j' + \gamma_i, \qquad v_{ji} = \alpha_i' x_j + \beta_i' y_j + \gamma_i'. \tag{3.3}$$

Now we define the "polygon-polygon" phi-function as

$$\Phi^{KK'} = \max \left\{ \max_{1 \leq i \leq m} \min_{1 \leq j \leq m'} u_{ij}, \ \max_{1 \leq j \leq m'} \min_{1 \leq i \leq m} v_{ji} \right\}, \tag{3.4}$$

see [14] for a detailed analysis of this formula. We note that $\Phi^{KK'}$ does not involve quadratic functions. It is defined by *linear* expressions only if rotational angles are not used.

In particular, if P is a half-plane defined by $\alpha x + \beta y + \gamma \leq 0$ and K' a polygon with vertices (x_i', y_i'), then (3.4) takes a much simpler form

$$\Phi^{PK'} = \min_{1 \leq j \leq m'} \alpha x_j' + \beta y_j' + \gamma. \tag{3.5}$$

3.1.1. Convex Polygon and Circle

Let K be a convex polygon with sides E_i and vertices (x_i, y_i) for $1 \leq i \leq m$. Let $\alpha_i x + \beta_i y + \gamma_i = 0$ be the equation of the line containing the side E_i. We assume that $\alpha_i^2 + \beta_i^2 = 1$ and the vertices and sides are numbered counterclockwise and the ith side joins the ith and $(i+1)$st vertices (if $i = m$, then we set $i + 1 = 1$). Let C be a circle with center (x_C, y_C) and radius r_C. Then we define

$$\Phi^{KC} = \max_{1 \leq i \leq m} \max\{\chi_i, \psi_i\}, \quad \psi_i = \min\{\omega_i, \mu_i\}, \tag{3.6}$$

where $\chi_i = \alpha_i x_C + \beta_i y_C + \gamma_i - r_C$ (see Figure 3(a)), $\omega_i = (x_i - x_C)^2 + (y_i - y_C)^2 - r_C^2$ (see Figure 3(b)), and $\mu_i = (\alpha_{i-1} - \alpha_i)(y_i - y_C) - (\beta_{i-1} - \beta_i)(x_i - x_C) + r_C(\alpha_{i-1}\beta_i - \alpha_i\beta_{i-1})$ (see Figure 3(b)). See [14] for more details.

In particular, if P is a half-plane $\alpha x + \beta y + \gamma \leq 0$ and C a circle with center (x_C, y_C) and radius r_C, then (3.6) takes a much simpler form

$$\Phi^{PC} = \alpha x_C + \beta y_C + \gamma - r_C. \tag{3.7}$$

3.2. Convex Polygon and Circular Segment

Let K be again a convex polygon. Let D be a circular segment $D = C \cap T$, where C is a circle and T a triangle made by the chord (the base of the segment) and the two tangents drawn at its endpoints. Now we define

$$\Phi^{KD} = \max\left\{\Phi^{KC}, \Phi^{KT}\right\}, \tag{3.8}$$

where Φ^{KC} was defined by (3.6) and Φ^{KT} by (3.4).

3.3. Two Circular Segments

Let $D = C \cap T$ and $D' = C' \cap T'$ be two circular segments. We define

$$\Phi^{DD'} = \max\left\{\Phi^{CC'}, \Phi^{TC'}, \Phi^{T'C}, \Phi^{TT'}\right\}, \tag{3.9}$$

where $\Phi^{CC'}$ was defined by (2.9), $\Phi^{TC'}$ and $\Phi^{T'C}$ by (3.6), and $\Phi^{TT'}$ by (3.4).
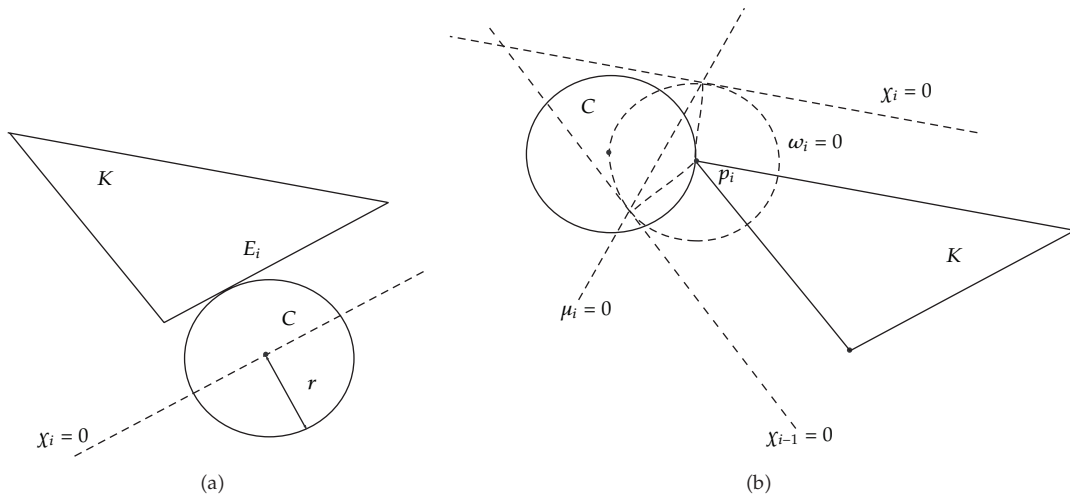
Figure 3: A convex polygon and a circle.

This takes care of all possible pairs of convex basic objects, that is, types (a) and (b). It remains to deal with concave objects, that is, "hats" and "horns." We first consider a simple object with a concave arc—the complement to a circle. This case is practically important because in many applications one places objects into a circular container.

3.3.1. Convex Objects inside a Circular Container

Let C^* denote the (closure of the) complement to a circle C with center (x_C, y_C) and radius r_C. Now let C' be a circular object with center $(x_{C'}, y_{C'})$ and radius $r_{C'} \leq r_C$ that we want to place inside the circle C. Then we define

$$\Phi^{C^*C} = (r_C - r_{C'})^2 - (x_C - x_{C'})^2 - (y_C - y_{C'})^2. \tag{3.10}$$

If $r_{C'} > r_C$, then we set $\Phi^{C^*C} = -\infty$.

Next let K be a polygon (not necessarily convex) with vertices $(x_1, y_1), \ldots, (x_m, y_m)$ that we want to place in our circle C. Then we set

$$\Phi^{C^*K} = \min_{1 \leq i \leq m} \left\{ r_C^2 - (x_i - x_C)^2 - (y_i - y_C)^2 \right\}. \tag{3.11}$$

If $H = T \cap C_1^*$ is a "hat," that is, the intersection of a triangle T and the complement to a circle C_1 (see Figure 6), we simply put $\Phi^{C^*H} = \Phi^{C^*T}$, where Φ^{C^*T} is given by (3.11).

Now let $D = C' \cap T$ be a circular segment, where C' is a circle with center $(x_{C'}, y_{C'})$ and radius $r_{C'}$ and T a triangle as before, and $q_i = (x_i, y_i)$, $i = 1, 2$, the endpoints of the chord bounding D; see Figure 4(a). We put

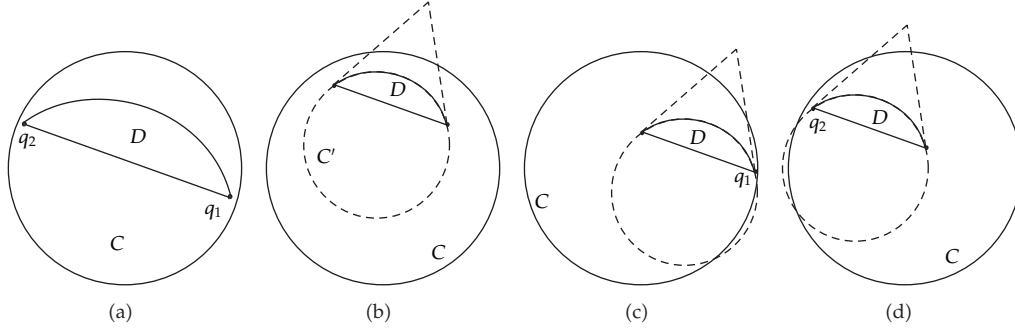$$\psi_0 = \min_{i=1,2} \left\{ r_C^2 - (x_i - x_C)^2 - (y_i - y_C)^2 \right\}, \tag{3.12}$$

Figure 4: Nonoverlapping C^* and D.

then

$$\Phi^{C^*D} = \psi_0 \quad \text{(for } r_C \leq r_{C'}\text{).} \tag{3.13}$$

Note that $\psi_0 = \Phi^{C^*K}$ subject to $m = 2$, see Figure 4(a). Now we define

$$\Phi^{C^*D} = \min\left\{\psi_0, \max\left\{\Phi^{C^*C'}, \varphi_1, \varphi_2\right\}\right\} \quad \text{(for } r_C > r_{C'}\text{),} \tag{3.14}$$

where ψ_0 is given by (3.12) and

$$\varphi_1 = (y_1 - y_{C'})(x_{C'} - x_C) - (x_1 - x_{C'})(y_{C'} - y_C),$$
$$\varphi_2 = -(y_2 - y_{C'})(x_{C'} - x_C) + (x_2 - x_{C'})(y_{C'} - y_C). \tag{3.15}$$

Formula (3.14) results from the following observations: $\Phi^{C^*D} \geq 0$ if $\psi_0 \geq 0$ subject to $\Phi^{C^*C'} \geq 0$ (see Figure 4(b)), or $\varphi_1 \geq 0$ (Figure 4(c)), or $\varphi_2 \geq 0$ (Figure 4(d)).

To clarify the role of the functions φ_1 and φ_2 we introduce vectors $e = (x_C - x_{C'}, y_C - y_{C'})$, $a_1 = (x_1 - x_{C'}, y_1 - y_{C'})$, $a_2 = (x_2 - x_{C'}, y_2 - y_{C'})$, $e_1 = (-(y_1 - y_{C'}), x_1 - x_{C'})$, and $e_2 = (y_2 - y_{C'}, -(x_2 - x_{C'}))$, as shown in Figure 5(a). Note that $a_1 \perp e_1$ and $a_2 \perp e_2$. In these notations,

$$\varphi_1 = \langle e, e_1 \rangle, \qquad \varphi_2 = \langle e, e_2 \rangle. \tag{3.16}$$

We call φ_1 and φ_2 "switch" functions. Note that $\max\{\Phi^{C^*C'}, \varphi_1, \varphi_2\} < 0$ if $\Phi^{C^*C'} < 0$ and $\varphi_1 < 0$ and $\varphi_2 < 0$, see Figure 5(a). However, there exist tree cases, where $\Phi^{C^*C'} < 0$ but $\max\{\Phi^{C^*C'}, \varphi_1, \varphi_2\} \geq 0$. First, $\varphi_1 \geq 0$ and $\varphi_2 < 0$ (see Figure 5(b)). Second, $\varphi_2 \geq 0$ and $\varphi_1 < 0$ (Figure 5(c)). Lastly, $\varphi_1 \geq 0$ and $\varphi_2 \geq 0$ (Figure 5(d)).

3.4. Polygon and Hat

Let $H = T \cap C^*$ be a hat, that is, the intersection of the complement to a circle C and a triangle T as shown in Figure 6. Let G denote the domain lying above the circle C and above the line
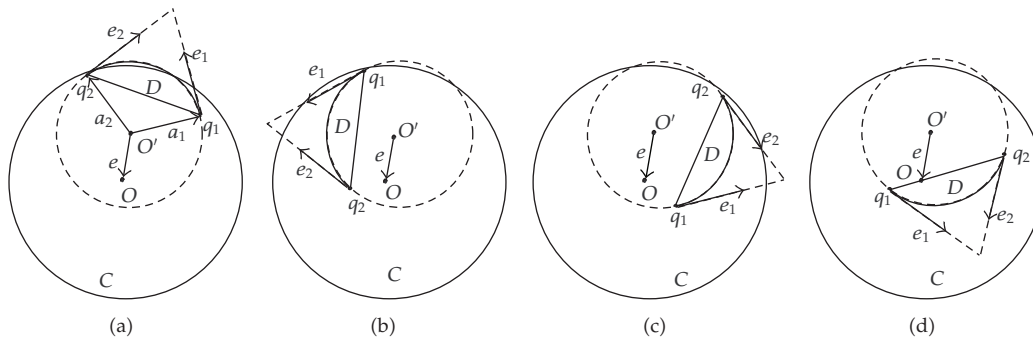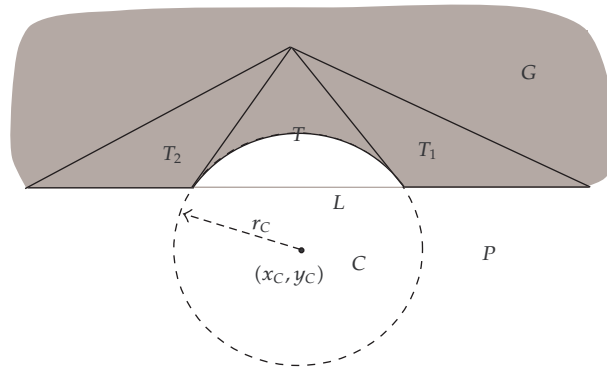
Figure 5: Role of the functions φ_1, φ_2.

Figure 6: The domain G (grey) and three triangles T, T_1, and T_2.

L containing the chord forming the base of the triangle T; see the grey area in Figure 6. Note that $H = G \cap T$.

Now if B is any *convex* object, then it overlaps with H if and only if it overlaps with both T and G; hence, we can define the phi-function as

$$\Phi^{HB} = \max\left\{\Phi^{TB}, \Phi^{GB}\right\} \tag{3.17}$$

provided we have properly defined Φ^{TB} and Φ^{GB}. This formula applies when B is either a convex polygon or a circular segment. In these two cases Φ^{TB} is given by either (3.4) or (3.8), respectively. Thus it remains to define function Φ^{GB}. Here we assume that B is a convex polygon.

Let T_1 and T_2 denote two triangles adjacent to T; one side of each is a tangent to the circle C, and another side of each is a segment of the line L adjacent to the chord; see Figure 6 (the choice of the third side is not important). Let the circle C have center (x_C, y_C) and radius r_C. Let the half-plane P below the line L be defined by inequality $\alpha_P x + \beta_P y + \gamma_P \geq 0$. Note that $G = (C \cup P)^* = C^* \cap P^*$.

Now let K be any polygon (not necessarily convex) with vertices $(x_1, y_1), \ldots, (x_m, y_m)$. Note that K does not overlap with G if and only if two conditions are met: (i) every vertex (x_i, y_i) lies either in the circle C or below the line L; (ii) the polygon K does not overlap with T_1 and T_2. Accordingly, we define

$$\Phi^{GK} = \min\left\{\Phi^{T_1 K}, \Phi^{T_2 K}, \Psi\right\}, \tag{3.18}$$

where

$$\Psi = \min_{1 \leq i \leq m} \max\left\{r_C^2 - (x_C - x_i)^2 - (y_C - y_i)^2, \alpha_P x_i + \beta_P y_i + \gamma_P\right\}. \tag{3.19}$$

Therefore,

$$\Phi^{HK} = \max\left\{\Phi^{TK}, \Phi^{GK}\right\}, \tag{3.20}$$

which completes the analysis of the "polygon-hat" pair.

3.5. Circular Segment and Hat

Let $H = T \cap C^* = T \cap G$ be a hat as before and D a circular segment. The above analysis applies, up to the formula (3.17), because D is a convex object. It remains to define Φ^{GD}.

We again use the notations P, C, and so forth, for objects associated with the hat H, as defined above. We denote by $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ the endpoints of the arc bounding H, as shown in Figure 8, that is, the points of intersection of ∂C with $\partial P = L$. The point $p_i = (x_i, y_i)$ is a vertex of the triangle T_i for $i = 1, 2$.

The circular segment $D = C' \cap T'$ is the intersection of a circle C' and a triangle T', as before. Let $r_{C'}$ denote the radius of the circle C' and $(x_{C'}, y_{C'})$ its center. Let $q_1 = (x_1', y_1')$ and $q_2 = (x_2', y_2')$ denote the endpoints of the chord bounding D (as shown in Figure 4) and L' the line passing through these points. Let the half-plane P' below the line L' (away from D) be defined by inequality $\alpha' x + \beta' y + \gamma' \geq 0$. Note that $D = C' \cap (P')^*$.

If $r_{C'} > r_C$, then we set

$$\Phi^{GD} = \max\left\{\Phi^{P^* T'}, \Phi^{C^* D}, \Phi^{GC'}, \varphi_1, \varphi_2\right\}, \tag{3.21}$$

where $\Phi^{P^* T'}$ is defined by (3.5), $\Phi^{C^* D}$ by (3.14), and for $i = 1, 2$ we set

$$\varphi_i = \min\left\{\Phi^{T_{3-i} D}, r_C^2 - (x_i' - x_C)^2 - (y_i' - y_C)^2, \right.$$
$$\left. \alpha' x_i + \beta' y_i + \gamma', -(\alpha' x_{3-i} + \beta' y_{3-i} + \gamma')\right\}. \tag{3.22}$$

Thus, $\Phi^{GD} \geq 0$ if $\Phi^{P^* T'} \geq 0$ (see Figure 7(a)) or $\Phi^{C^* D} \geq 0$ (see Figure 7(b)) or $\Phi^{GC'} \geq 0$ (see Figure 7(c)), $\varphi_1 \geq 0$ (see Figure 7(d)) or $\varphi_2 \geq 0$ (see Figure 7(e)).
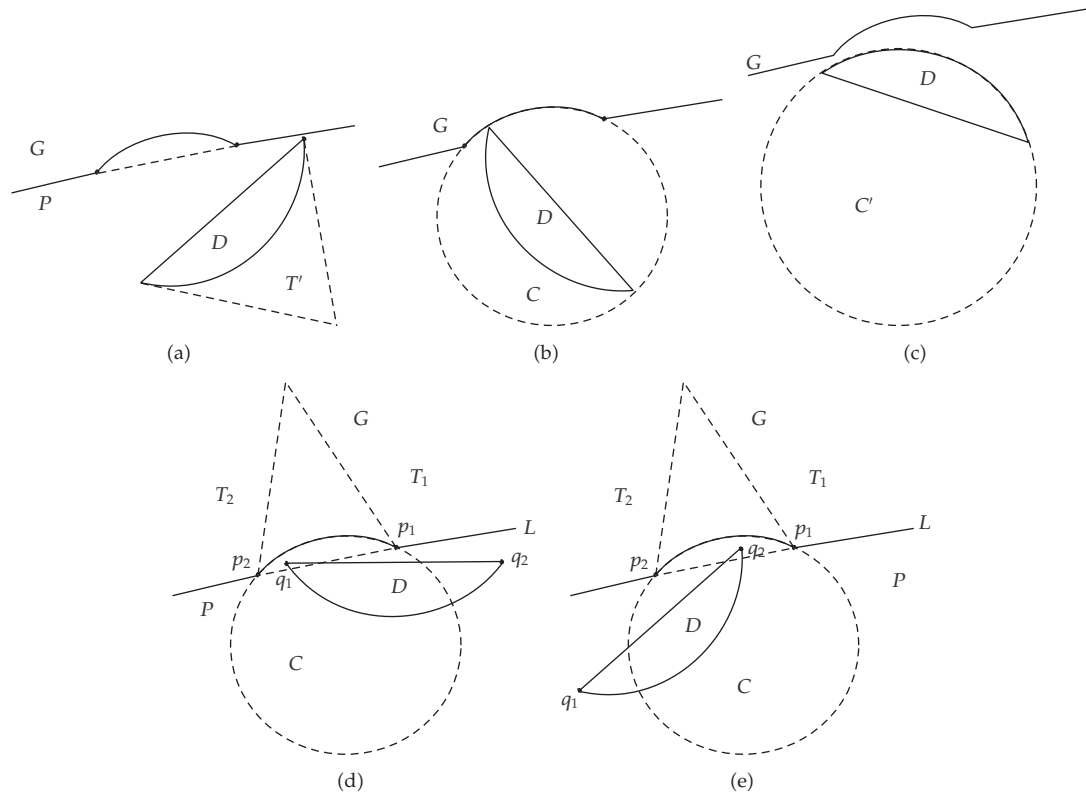
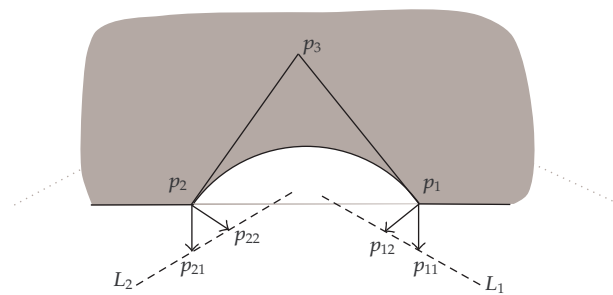**Figure 7:** A circular segment D versus the region G.

Figure 8: The lines L_1 and L_2 for a segment-hat pair.

The function $\Phi^{GC'}$ in (3.21) is defined as follows:

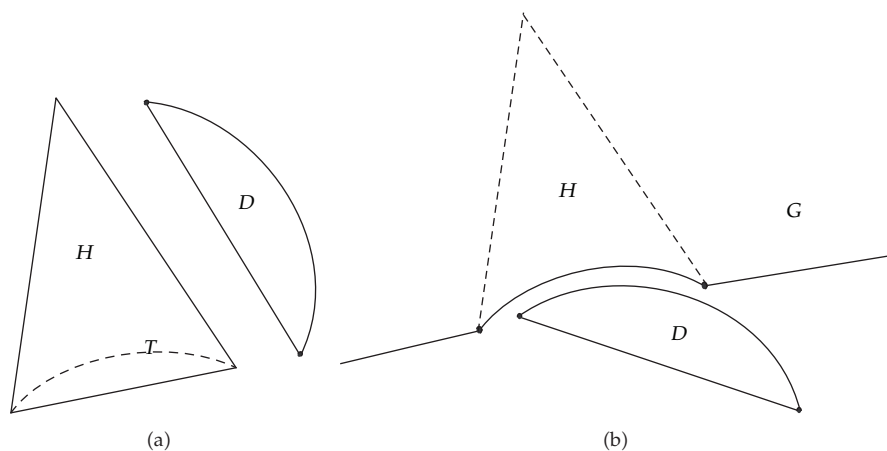$$\Phi^{GC'} = \max\left\{\Phi^{C^*C'}, \Phi^{P^*C'}, \min\{\omega_1, \psi_1, \omega_2, \psi_2\}\right\}, \tag{3.23}$$

Figure 9: A hat H and a circular segment D.

where $\Phi^{C^*C'}$ was defined by (3.10) and $\Phi^{P^*C'}$ by (3.7). We also denote

$$\omega_i = (x_i - x_{C'})^2 + (y_i - y_{C'})^2 - r_{C'}^2 \tag{3.24}$$

for $i = 1, 2$, and the functions ψ_1, ψ_2 are defined so that $\psi_i = 0$ is the equation of the line L_i (see Figure 8), and $\psi_i \geq 0$ is the half-plane below that line in Figure 8. The line L_i passes through points p_{i1} and p_{i2}. The line segment $p_i p_{i1}$, $i = 1, 2$, is perpendicular to the line $p_1 p_2$, the line segment $p_i p_{i2}$ is perpendicular to the line $p_i p_3$, and we have $\|p_i p_{ij}\| = r_{C'}$ for all $i, j = 1, 2$. The functions ψ_i, $i = 1, 2$, come from the application of (3.6) to T_i and C'.

If $r_{C'} \leq r_C$, we need to replace φ_i, $i = 1, 2$, in (3.21) with φ'_i defined by

$$\varphi'_i = \min\{\varphi_i, \alpha'(x_{C'} - x_C) + \beta'(y_{C'} - y_C)\}. \tag{3.25}$$

Finally, combining (3.17) and (3.21) gives

$$\Phi^{HD} = \max\{\Phi^{TD}, \Phi^{GD}\}. \tag{3.26}$$

Indeed, $\Phi^{HD} \geq 0$ if $\Phi^{TD} \geq 0$ (see Figure 9(a)) or $\Phi^{GD} \geq 0$ (see Figure 9(b)).

3.6. Two Hats

Let $H' = G' \cap T'$ be a hat and $H'' = G'' \cap T''$ another hat. Equivalently, $H' = (C')^* \cap T'$ and $H'' = (C'')^* \cap T''$. For the hat H' we use notation G', C', T', and so forth, as defined above, and for the hat H'' the respective notation G'', C'', T'', and so forth. Now our phi-function is defined by

$$\Phi^{H'H''} = \max\{\Phi^{T'H''}, \Phi^{G'T''}, \omega, \tau\}, \tag{3.27}$$
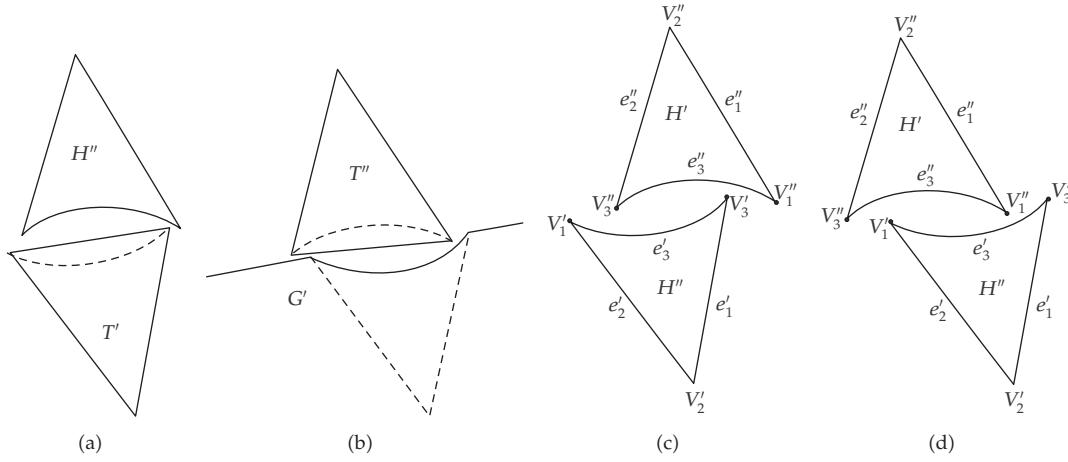
Figure 10: Four cases of nonoverlapping hats H' and H''.

where $\Phi^{T'H''}$, $\Phi^{G'T''}$ are given by (3.17), (3.18), respectively, and we denote

$$
\begin{aligned}
\omega = \min\Big\{ &\alpha_2' x_1'' + \beta_2' y_1'' + \gamma_2', \alpha_2'' x_1' + \beta_2'' y_1' + \gamma_2'', \\
&(r_{C'})^2 - (x_{C'} - x_3'')^2 - (y_{C'} - y_3'')^2, \\
&(r_{C''})^2 - (x_{C''} - x_3')^2 - (y_{C''} - y_3')^2 \Big\}, \\
\tau = \min\Big\{ &\alpha_1' x_3'' + \beta_1' y_3'' + \gamma_1', \alpha_1'' x_3' + \beta_1'' y_3' + \gamma_1'', \\
&(r_{C'})^2 - (x_{C'} - x_1'')^2 - (y_{C'} - y_1'')^2, \\
&(r_{C''})^2 - (x_{C''} - x_1')^2 - (y_{C''} - y_1')^2 \Big\},
\end{aligned}
\tag{3.28}
$$

where (x_i', y_i') are the coordinates of the vertices and $\alpha_i' x + \beta_i' y + \gamma_i' = 0$, $i = 1, 2$, are the equations of lines containing the two straight sides of H', respectively; $(x_{C'}, y_{C'})$ and $r_{C'}$ are the coordinates of the center and the radius of the arc bounding H'. Similar notation applies to the hat H''.

The hats H' and H'' do not overlap if $\Phi^{T'H''} \geq 0$ (see Figure 10(a)), or $\Phi^{G'T''} \geq 0$ (Figure 10(b)), or $\omega \geq 0$ (Figure 10(c)), or $\tau \geq 0$ (Figure 10(d)).

3.7. Horns

A horn $V = H \cap (D \cup T)$ is the intersection of a hat H and the union of a circular segment D and a triangle T; see Figure 11, where the triangle T has vertices p_1, p_2, p_3, and the hat H has vertices p_1, p_2, p_4.

Now for any convex polygon K we define

$$
\Phi^{VK} = \max\Big\{ \Phi^{HK}, \min\big\{ \Phi^{KD}, \Phi^{KT} \big\} \Big\}.
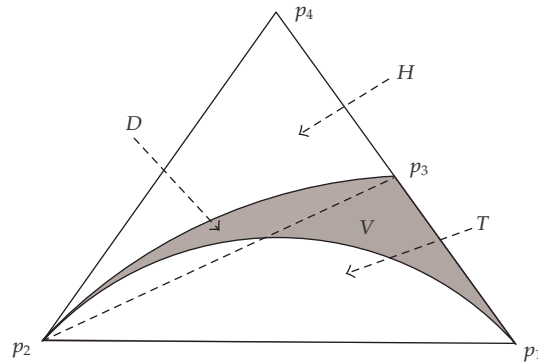\tag{3.29}
$$

Figure 11: A horn V (grey) and the respective hat H (with vertices p_1, p_2, p_4), circular segment D, and triangle T (with vertices p_1, p_2, p_3).

Similarly, for any circular segment D',

$$\Phi^{VD'} = \max\left\{\Phi^{HD'}, \min\left\{\Phi^{DD'}, \Phi^{TD'}\right\}\right\} \tag{3.30}$$

and for any hat H'

$$\Phi^{VH'} = \max\left\{\Phi^{HH'}, \min\left\{\Phi^{H'D}, \Phi^{H'T}\right\}\right\}. \tag{3.31}$$

Now let $V' = H' \cap (D' \cup T')$ be another horn. Then we define

$$\Phi^{VV'} = \max\left\{\Phi^{HH'}, \min\left\{\Phi^{HD'}, \Phi^{HT'}\right\}, \min\left\{\Phi^{H'D}, \Phi^{H'T}\right\}, \min\left\{\Phi^{DD'}, \Phi^{TT'}, \Phi^{TD'}, \Phi^{T'D}\right\}\right\}. \tag{3.32}$$

Some formulas for the phi-functions may appear quite complex. Note, however, that they all can be programmed off-line and stored in a computer library. In practical applications, one can just call the respective functions, and their evaluation proves to be fast and efficient.

4. Adjusted Phi-Functions

Some applications involve restrictions on the distances between certain pairs of objects, or between objects and the walls of the container. For example, when one is packing radioactive waste, discarded pieces cannot be placed too close together. On the other hand, when one designs a printed circuit board (PCB), then certain electronic components cannot be placed too far apart. Cutting mechanical parts out of a metal sheet is another example where minimal distances have to be maintained, because one has to take into account the physical size of the cutter.

In other words, some upper and/or lower limits on the distances between certain placed objects may be set, that is, given two objects A, B, the corresponding placed objects $A' = A(v_A, \theta_A)$ and $B' = B(v_B, \theta_B)$ must satisfy

$$\text{dist}(A', B') \geq \rho_{AB}^- \quad \text{or} \quad \text{dist}(A', B') \leq \rho_{AB}^+, \tag{4.1}$$

where

$$\text{dist}(A', B') = \min_{X \in A', \, Y \in B'} \text{dist}(X, Y). \tag{4.2}$$

Here ρ_{AB}^- denotes the minimal allowable distance and ρ_{AB}^+ the maximal allowable distance between A' and B'.

To fulfil (4.1), it has been a common practice to compute the actual distance between A' and B' at every step during the optimization process and check if (4.1) holds. But the computation of geometric distances (especially for complex objects) involves complicated formulas with radicals, see a variety of examples detailed in [11]. We avoid the computation of geometric distances by using so called *adjusted* phi-functions defined below.

Suppose we have to maintain a minimal distance $\rho = \rho_{AB}^-$ for a pair of objects A, B. We will construct an *adjusted* phi-function $\widehat{\Phi}^{AB}$ satisfying

$$\begin{aligned}
\widehat{\Phi}^{AB} &> 0 \quad \text{if } \text{dist}(A', B') > \rho, \\
\widehat{\Phi}^{AB} &= 0 \quad \text{if } \text{dist}(A', B') = \rho, \\
\widehat{\Phi}^{AB} &< 0 \quad \text{if } \text{dist}(A', B') < \rho.
\end{aligned} \tag{4.3}$$

Then we work with it a just like with the regular phi-function Φ^{AB} in the previous sections, where no restrictions on distances were imposed. Indeed, all allowable placements of the objects A, B now correspond to $\widehat{\Phi}^{AB} \geq 0$ and prohibited placements correspond to $\widehat{\Phi}^{AB} < 0$. Thus our optimization algorithms can proceed the usual routine, but with the new (adjusted) phi-function $\widehat{\Phi}^{AB}$ instead of Φ^{AB}.

Given an object A and $\rho > 0$ we define its *ρ-expansion* (Figure 12) by

$$\widehat{A} = \widehat{A}_\rho = A \oplus C_\rho, \tag{4.4}$$

where (C, ρ) denotes a circle of radius ρ centered on the origin and the symbol $\oplus$ stands for the so-called Minkowski sum [19], which is defined by

$$A_1 \oplus A_2 = \left\{ (x_1 + x_2, y_1 + y_2) \colon (x_1, y_1) \in A_1, \ (x_2, y_2) \in A_2 \right\} \tag{4.5}$$

for any two sets $A_1, A_2 \subset \mathbb{R}^2$. In other words, the ρ-expanded object $\widehat{A}$ in (4.4) consists of points that are either *in A* or *at distance $\leq \rho$ from A*. We will *not* need to use Minkowski sum for computing our phi-functions.

Now we construct the adjusted phi-function by $\widehat{\Phi}^{AB} = \Phi^{\widehat{A}B}$, and it will satisfy the requirements (4.3). Note that instead of expanding the object A we can expand the other
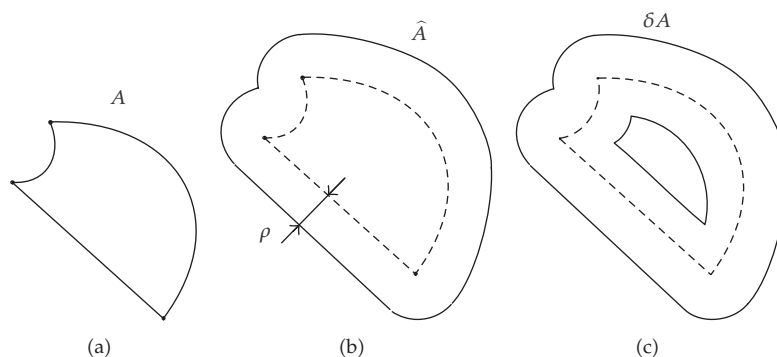
Figure 12: An object A (a), its ρ-expansion $\hat{A}$ (b), and the ρ-expansion δA of its boundary ∂A (c).

object B and define $\hat{\Phi}^{AB} = \Phi^{A\hat{B}}$. This extra flexibility can be used in practice to minimize the cost of computation.

Suppose we have to maintain a maximal allowable distance $\rho = \rho^+$ for a pair of objects A, B. This means that the objects have to be positioned so that $\Phi^{AB} \geq 0$ (to avoid overlaps) and $\hat{\Phi}^{AB} \leq 0$, where $\hat{\Phi}^{AB}$ is the adjusted function constructed above (the latter condition will keep the distance $\leq \rho$). Thus we can define another adjusted phi-function as

$$\check{\Phi}^{AB} = \min\left\{\Phi^{AB}, -\hat{\Phi}^{AB}\right\}. \tag{4.6}$$

Now we have

$$\check{\Phi}^{AB} > 0 \quad \text{if } 0 < \text{dist}(A', B') < \rho,$$

$$\check{\Phi}^{AB} = 0 \quad \text{if } (\text{int}(A') \cap \text{int}(B') = \emptyset \text{ and } \partial A' \cap \partial B' \neq \emptyset) \text{ or } \text{dist}(A', B') = \rho, \tag{4.7}$$

$$\check{\Phi}^{AB} < 0 \quad \text{if } \text{int}(A') \cap \text{int}(B') \neq \emptyset \text{ or } \text{dist}(A', B') > \rho.$$

Thus all allowable positions of A and B correspond to $\check{\Phi}^{AB} \geq 0$.

We see that the adjusted phi-functions can always be defined as (ordinary) phi-functions, but for expanded objects. It remains to define phi-functions for expanded objects. For any object A we have $\hat{A} = A \cup \delta A$, where $\delta A = \partial A \oplus C_\rho$ is the expansion of the boundary of A; see Figure 12(b). One can think of δA as a "fattened" boundary of A (whose "thickness" is 2ρ). Then by the decomposition principle (2.11) we define

$$\hat{\Phi}^{AB} = \Phi^{\hat{A}B} = \min\left\{\Phi^{AB}, \Phi^{\delta AB}\right\}. \tag{4.8}$$

Now recall that ∂A consists of linear segments and circular arcs, that is, $\partial A = \cup_{i=1}^{m} \gamma_i$, where each γ_i is either a segment of a line or a circular arc. Therefore, $\delta A = \cup_{i=1}^{m} \hat{\gamma}_i$, where
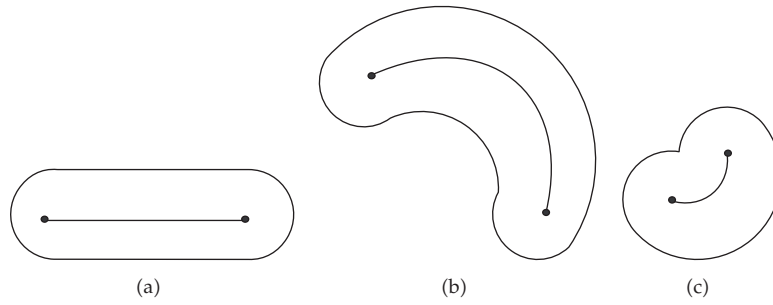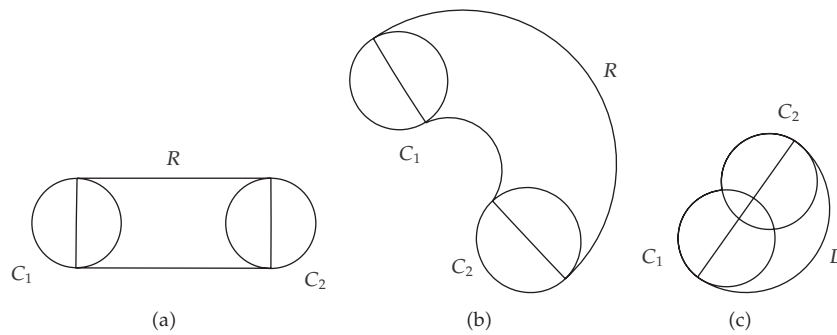
Figure 13: Expansion of boundary components.

Figure 14: Various domains $\widehat{\gamma} = C_1 \cup C_2 \cup R$.

$\widehat{\gamma}_i = \gamma_i \oplus C_\rho$ denotes the expansion of γ_i (as described below). Now by the decomposition principle (2.11) we define

$$\Phi^{\delta AB} = \min_{1 \leq i \leq m} \Phi^{\widehat{\gamma}_i B}. \tag{4.9}$$

The domain $\widehat{\gamma}_i = \gamma_i \oplus C_\rho$ is shown in Figure 13 for three different cases: γ_i is a line segment (a), γ_i is an arc of radius $> \rho$ (b), and γ_i is an arc of radius $\leq \rho$ (c); see a detailed analysis below.

In all these three cases we have $\widehat{\gamma}_i = C_1 \cup C_2 \cup R$, where C_1 and C_2 are disks of radius ρ centered on the endpoints of γ_i. If γ_i is a line segment, then R is a rectangle (Figure 14(a)). If γ_i is a circular arc of radius $r_i > \rho$, then R is a "bent rectangle" (Figure 14(b)). If γ_i is a circular arc of radius $r_i \leq \rho$, then R degenerates to a circular segment (Figure 14(c)). Thus we define

$$\Phi^{\widehat{\gamma}_i B} = \min\left\{\Phi^{C_1 B}, \Phi^{C_2 B}, \Phi^{RB}\right\}, \tag{4.10}$$

where R denotes the corresponding rectangle, or bent rectangle, or circular segment.

We note that ∂A consists of m components, so δA will consist of m disks of radius ρ and m rectangles or "bent rectangles" (some of the latter may degenerate to circular segments). Rectangles and circular segments are objects of basic types, for which phi-functions were defined in Section 3. Bent rectangles are objects of a new type, so we need to handle them separately.
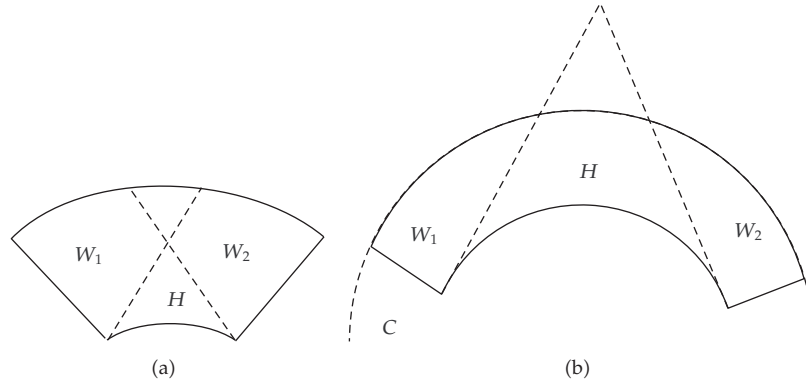
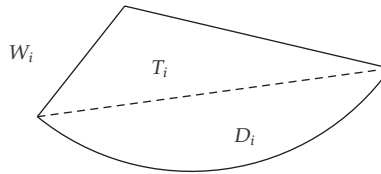**Figure 15:** A "bent rectangle" $R = W_1 \cup W_2 \cup H$ (a) and $R = W_1 \cup W_2 \cup (H \cap C)$ (b).

Figure 16: A wedge $W_i = T_i \cup D_i$.

We have two cases shown in Figure 15. On (a), the "bent rectangle" is the union of two "wedges" W_1, W_2 and a hat H. Every wedge W_i is in turn the union of a triangle T_i and a circular segment D_i (see Figure 16), hence

$$\Phi^{RB} = \min\left\{\Phi^{HB}, \Phi^{T_1 B}, \Phi^{D_1 B}, \Phi^{T_2 B}, \Phi^{D_2 B}\right\}. \tag{4.11}$$

In the second case (Figure 15(b)) the bent rectangle R can be decomposed as $R = W_1 \cup W_2 \cup (H \cap C)$. Accordingly, we define

$$\Phi^{RB} = \min\left\{\Phi^{W_1 B}, \Phi^{W_2 B}, \max\left\{\Phi^{HB}, \Phi^{CB}\right\}\right\}. \tag{4.12}$$

5. Numerical Examples

We illustrate our method by several model examples. In these examples we describe each object by listing elements of its boundary $\partial A = \{l_1, \ldots, l_n\}$. Each boundary element l_i is completely described by its numerical code (which is 0 for straight line segments, +1 for convex arcs, and −1 for concave arcs), the coordinates of the two endpoints (x_1, y_1) and (x_2, y_2), and (for circular arcs only) the coordinates of the center (x_C, y_C).

Our goal is to place a given object or two given objects into a circle of minimal radius or into a rectangle of minimal area. The rectangle is always properly oriented, that is, its sides are parallel to the x and y axes. Accordingly, our objective function (to be minimized) is
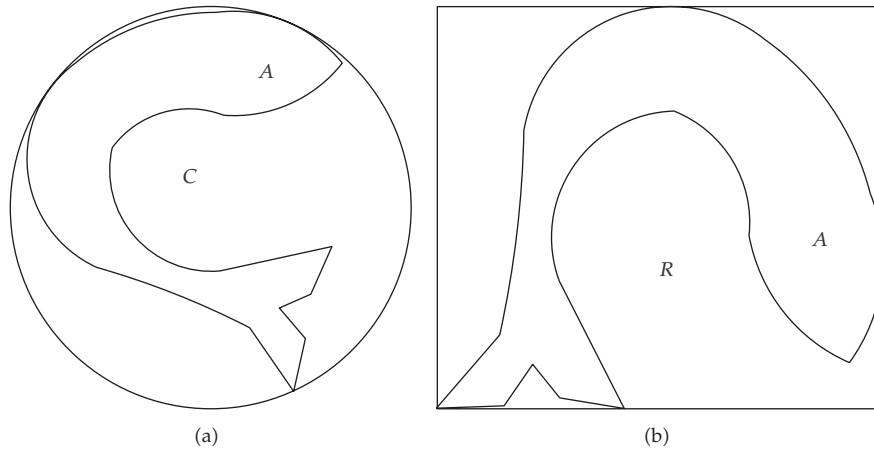
(a)   (b)

Figure 17: A dolphin-like object placed into (a) a circle of minimal radius and (b) a rectangle of minimal area.
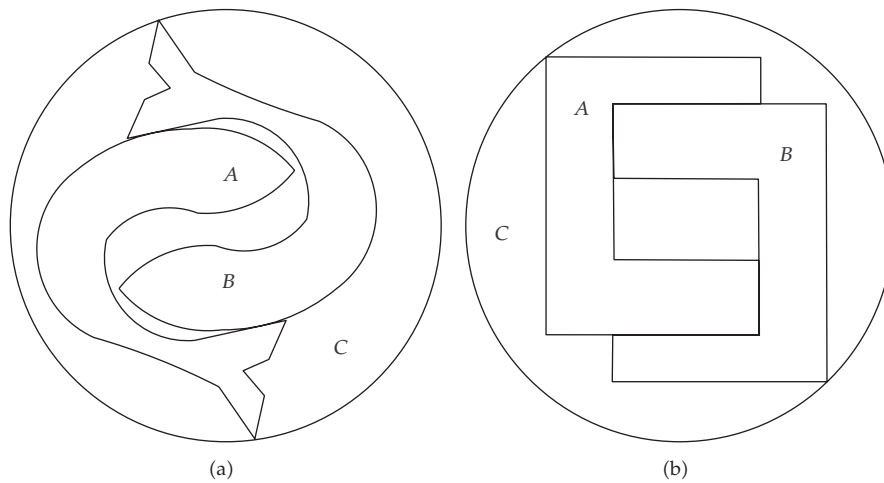

(a)   (b)

Figure 18: (a) Two dolphin-like objects placed into a circle of minimal radius and (b) two staple-like objects placed into a circle of minimal radius.

$F(u_1, u_2, \ldots) = r$ in case of a circular container of radius r and $F(u_1, u_2, \ldots) = ab$ in case of a rectangular container with sides a and b.

The arguments $u_1, u_2, \ldots$ of the objective function include the translation vectors $v = (v_1, v_2)$ for all the objects and rotation angles θ, where appropriate; compare (2.1). If we place a single object into a circle, no rotation is needed. If two objects are placed into a circle, it is enough to rotate one of them to achieve the optimal placement. When one or two objects are placed into a rectangle, each of them may have to be rotated in order to find the best placement.

Example 5.1. The goal is to place a given object into a circle of minimal radius. The object is a dolphin-like domain A shown in Figure 17(a); its boundary is described in Table 1.

Table 1: The boundary of the dolphin-like has 13 elements.

Code	x_1, y_1	x_C, y_C	x_2, y_2
1	3.753993, 2.279451	1.586533, 0.524744	1.233706, 3.291040
1	1.233706, 3.291040	1.197359, −1.021342	−1.57802, 2.279451
1	−1.578020, 2.279451	−0.164213, 0.362623	−1.15153, −1.804929
−1	−1.151532, −1.804929	−5.628434, −17.641177	1.905534, −3.010089
0	1.905534, −3.010089		2.781521, −4.283507
0	2.781521, −4.283507		3.018245, −3.228149
0	3.018245, −3.228149		2.493878, −2.619361
0	2.493878, −2.619361		3.122744, −2.338969
0	3.122744, −2.338969		3.546228, −1.389653
0	3.546228, −1.389653		1.313080, −1.873368
−1	1.313080, −1.873368	1.127739, 0.139935	−0.84345, 0.589527
−1	−0.843455, 0.589527	0.696106, −0.550254	1.384174, 1.237457
1	1.384174, 1.237457	1.595781, 3.972047	3.753993, 2.279451

Table 2: The boundary of the "thick staple" object has 8 elements, all straight line segments.

Code	x_1, y_1	x_C, y_C	x_2, y_2
0	1.182742, 1.708476		1.278823, 1.694810
0	−1.278823, 1.694810		−1.278823, −1.482497
0	−1.278823, −1.482497		1.175904, −1.482497
0	1.175904, −1.482497		1.175904, −0.635215
0	1.175904, −0.635215		−0.499328, −0.635215
0	−0.499328, −0.635215		−0.499328, 1.175508
0	−0.499328, 1.175508		1.196417, 1.175508
0	1.196417, 1.175508		1.182742, 1.708476

The optimal placement is also shown in Figure 17(a). The radius of the optimal circle is $r^* = 4.015234$. This example took 3.61 sec of the computer running time (we processed our examples on a PC with an AMD Athlon 64X2 2.6 GHz CPU).

Example 5.2. The goal is to place the given object (same as in Example 5.1) into a rectangle of minimal area. The optimal placement is shown in Figure 17(b). The rectangle has sides $a^* = 7.132090$ and $b^* = 6.416804$. We note that our algorithm supports rotation of objects. The optimal rectangle is found when the object A is rotated by angle $\theta_A = 1.31245$. This example took 104 sec.

Example 5.3. The goal is to place two given objects, A and B, into a circle of minimal radius. The objects are identical copies of the dolphin-like object in Example 5.1. The optimal placement is shown in Figure 18(a). The radius of the circle is $r^* = 5.251253$. Again, the objects are subject to rotation, and the optimal circle is found when the object B is rotated by angle $\theta_B = 3.141593$. This example took 4298 sec.

Table 3: The object A (Figure 19) bounded by three arcs.

Code	x_1, y_1	x_C, y_C	x_2, y_2
−1	−2.555404, −2.066713	0.651314, −15.372221	4.266788, −2.171920
1	4.266788, −2.171920	−0.740153, −1.703527	−2.19745, 3.109491
−1	−2.197449, 3.109491	−5.005667, 0.703211	−2.55540, −2.066713

Table 4: The object $B = H_1 \cup H_2$ (Figure 19) is the union of two overlapping hats. Each hat is specified by the coordinates of the two endpoints (x_1, y_1) and (x_2, y_2) and the center (x_C, y_C) of the circular arc bounding it and by the coordinates (x_v, y_v) of its third vertex. We note that both circular arcs bounding H_1 and H_2 have radius $r = 5.0$.

Hat	x_1, y_1	x_2, y_2	x_C, y_C	x_v, y_v
H_1	−2.878715, −1.116315	2.088654, 4.400499	−2.884564, 3.883681	2.661289, −1.109835
H_2	−1.442662, −4.818566	−1.442654, 2.605045	−4.792658, −1.106758	2.670029, −1.106765

Example 5.4. The goal is to place two given objects, A and B, into a circle of minimal radius. The objects have identical shape, they look like thick metal staples and their boundary is described in Table 2. The optimal placement is shown in Figure 18(b). The radius of the optimal circle is $r^* = 2.455866$. Again, the objects are subject to rotation, and the optimal circle is found when the object B is rotated by angle $\theta_B = 3.141593$. This example took 31 sec.

Example 5.5. The goal is to place two given objects, A and B, of *different shape* into a circle of minimal radius. The object A is made by three arcs (two concave and one convex) described in Table 3. The object $B = H_1 \cup H_2$ is the union of two overlapping hats specified in Table 4. The optimal placement is shown in Figure 19(a). The radius of the circle is $r^* = 5.322824$. Again, the objects are subject to rotation, and the optimal circle is found when the object B is rotated by angle $\theta_B = 2.309901$. This example took 1147 sec.

Example 5.6. The goal is to place the two given objects, A and B (same as in Example 5.5), into a rectangle of minimal area. The optimal placement is shown in Figure 19(b). The rectangle has sides $a^* = 13.294256$ and $b^* = 5.603828$. Again, the objects are subject to rotation, and the optimal rectangle is found when the object A is rotated by angle $\theta_A = -0.118376$ and the object B is rotated by angle $\theta_B = 0.715346$. This example took 443 sec.

Example 5.7. The goal is to place two very irregular (star-shaped) objects, A and B, into a circle of minimal radius. The objects have identical shape, each is the union of four overlapping hats specified in Table 5. The optimal placement is shown in Figure 20(a). The radius of the circle is $r^* = 7.031531$. Again, the objects are subject to rotation, and the optimal circle is found when the object B is rotated by angle $\theta_B = 0.634543$. This example took 996 sec.

Example 5.8. The goal is to place two very irregular (star-shaped) objects, A and B, into a rectangle of minimal area. The objects are the same as in Example 5.7. The optimal placement is shown in Figure 20(b). The rectangle has sides $a^* = 8.856350$ and $b^* = 14.292623$. Again, the objects are subject to rotation, and the optimal rectangle is found when the object A is rotated by angle $\theta_A = 0.470376$ and the object B is rotated by angle $\theta_B = 3.611969$. This example took 154 sec.
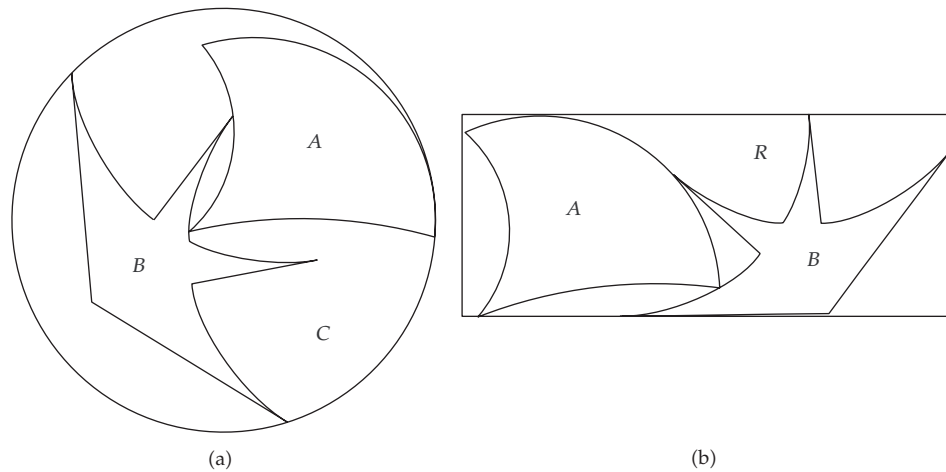
Figure 19: (a) Two objects of different shape placed into (a) a circle of minimal radius and (b) a rectangle of minimal area.
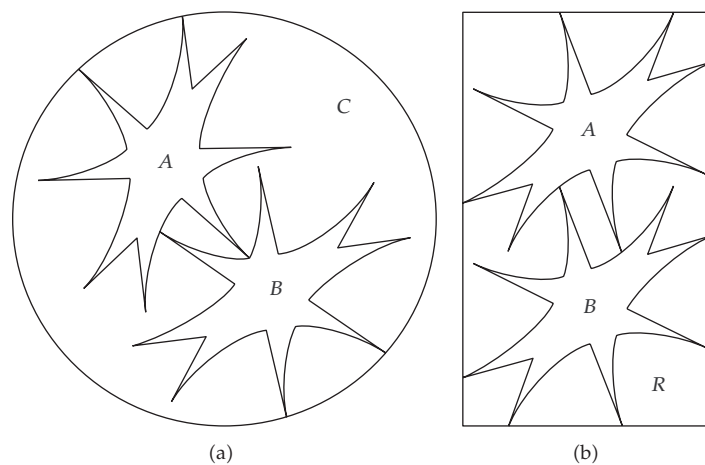

Figure 20: (a) Two objects of different shape placed into (a) a circle of minimal radius and (b) a rectangle of minimal area.

Example 5.9. This is a modification of Example 5.5: we place two objects A and B into a circle C of minimal radius, but now the object A must be at least the distance of 0.7 away from the object B and from the edge of the circle C; that is, we need dist$(A, B) \geq 0.7$ and dist$(A, C^*) \geq 0.7$. In this example we use adjusted phi-functions (Section 4). The optimal placement is shown in Figure 21. Note that the object A does not touch the object B or the boundary of C^*, to maintain the required distance from both. The radius of the optimal circle is $r^* = 5.823507$. The objects are subject to rotation, and the optimal circle is found when the object B is rotated by angle $\theta_B = 2.322388$. This example took 7725 sec.
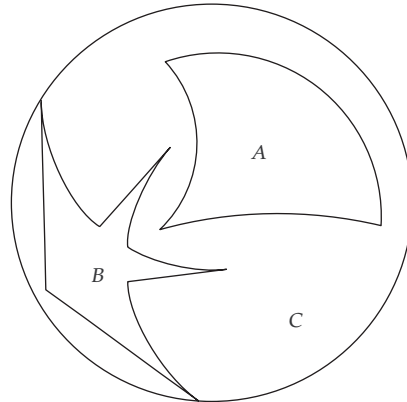
Figure 21: Two objects A and B placed into a circle C of minimal radius, with additional restriction on distances from A to B and C.

Table 5: The object $A = H_1 \cup H_2 \cup H_3 \cup H_4$ (Figure 20) is the union of four overlapping hats. Hats are specified as in Table 4. We note that all the circular arcs bounding these four hats have radius $r = 5.0$.

Hat	x_1, y_1	x_2, y_2	x_C, y_C	x_v, y_v
H_1	$-4.284755, -1.280657$	$0.682614, 4.236157$	$-4.290604, 3.719339$	$1.255249, -1.274177$
H_2	$-2.848702, -4.982908$	$-2.848694, 2.440703$	$-6.198698, -1.271100$	$1.263989, -1.271107$
H_3	$4.249704, -0.281005$	$-0.717661, -5.79782$	$4.255557, -5.281$	$-1.2903, -0.287489$
H_4	$2.813649, 3.421245$	$2.813646, -4.002366$	$6.163647, -0.29056$	$-1.29904, -0.290559$

Acknowledgments

References

[1] E. G. Birgin, J. M. Martínez, F. H. Nishihara, and D. P. Ronconi, "Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization," *Computers & Operations Research*, vol. 33, no. 12, pp. 3535–3548, 2006.

[2] A. M. Gomes and J. F. Oliveira, "Solving irregular strip packing problems by hybridising simulated annealing and linear programming," *European Journal of Operational Research*, vol. 171, pp. 811–829, 2006.

[3] V. J. Milenkovic and K. Daniels, "Translational polygon containment and minimal enclosure using mathematical programming," *International Transactions in Operational Research*, vol. 6, no. 5, pp. 525–554, 1999.

[4] J. A. Bennell and J. F. Oliveira, "The geometry of nesting problems: a tutorial," *European Journal of Operational Research*, vol. 184, no. 2, pp. 397–415, 2008.

[5] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *European Journal of Operational Research*, vol. 183, pp. 1109–1130, 2007.

[6] E. Burke, R. Hellier, G. Kendall, and G. Whitwell, "A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem," *Operations Research*, vol. 54, no. 3, pp. 587–601, 2006.

[7] V. Milenkovic and E. Sacks, "Two approximate Minkowski sum algorithms," *International Journal of Computational Geometry & Applications*, vol. 20, no. 4, pp. 485–509, 2010.

[8] V. J. Milenkovic, "Rotational polygon overlap minimization and compaction," *Computational Geometry*, vol. 10, no. 4, pp. 305–318, 1998.

[9] V. J. Milenkovic, "Rotational polygon containment and minimum enclosure using only robust 2D constructions," *Computational Geometry*, vol. 13, no. 1, pp. 3–19, 1999.

[10] E. Burke, R. Hellier, G. Kendall, and G. Whitwell, "Irregular packing using the line and arc no-fit polygon," *Operations Research*, vol. 58, pp. 948–970, 2010.

[11] J. Bennell, G. Scheithauer, Y. Stoyan, and T. Romanova, "Tools of mathematical modeling of arbitrary object packing problems," *Annals of Operations Research*, vol. 179, pp. 343–368, 2010.

[12] Y. Stoyan, G. Scheithauer, N. Gil, and T. Romanova, "Φ-functions for complex 2D-objects," *4OR. Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, vol. 2, no. 1, pp. 69–84, 2004.

[13] Yu. Stoyan, J. Terno, G. Scheithauer, N. Gil, and T. Romanova, "Phi-functions for primary 2D-objects," *Studia Informatica Universalis*, vol. 2, pp. 1–32.

[14] N. Chernov, Yu. Stoyan, and T. Romanova, "Mathematical model and efficient algorithms for object packing problem," *Computational Geometry*, vol. 43, no. 5, pp. 535–553, 2010.

[15] Y. G. Stoyan and A. Chugay, "Packing cylinders and rectangular parallelepipeds with distances between them," *European Journal of Operational Research*, vol. 197, pp. 446–455, 2008.

[16] T. Romanova, G. Scheithauer, and A. Krivulya, "Covering a polygonal region by rectangles," *Computational Optimization and Applications*, vol. 48, no. 3, pp. 675–695, 2011.

[17] http://www.math.uab.edu/~chernov/CP/.

[18] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[19] H. Minkovski, "Dichteste gitterförmige Lagerung kongruenter Körper," *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen*, pp. 311–355, 1904.