Journal of Global Optimization manuscript No.
(will be inserted by the editor)

Cutting Ellipses from Area-Minimizing Rectangles

Josef Kallrath · Steffen Rebennack

Received: 03/31/2013 / Accepted: 11/20/2013

Abstract A set of ellipses, with given semi-major and semi-minor axes, is to be cut from a rectangular design plate, while minimizing the area of the design rectangle. The design plate is subject to lower and upper bounds of its widths and lengths; the ellipses are free of any orientation restrictions. We present new mathematical programming formulations for this ellipse cutting problem. The key idea in the developed non-convex nonlinear programming models is to use separating hyperlines to ensure the ellipses do not overlap with each other. For small number of ellipses we compute feasible points which are globally optimal subject to the finite arithmetic of the global solvers at hand. However, for more than 14 ellipses none of the local or global NLP solvers available in `GAMS` can even compute a feasible point. Therefore, we develop polylithic approaches, in which the ellipses are added sequentially in a strip-packing fashion to the rectangle restricted in width, but unrestricted in length. The rectangle's area is minimized in each step in a greedy fashion. The sequence in which we add the ellipses is random; this adds some GRASP flavor to our approach. The polylithic algorithms allow us to compute good, near optimal solutions for up to 100 ellipses.

Keywords Global optimization · non-convex nonlinear programming · mixed integer programming · cutting stock problem · packing problem · shape

J. Kallrath
BASF SE
Scientific Computing
GVM/S-B009
D-67056 Ludwigshafen, Germany
E-mail: josef.kallrath@web.de

S. Rebennack
Colorado School of Mines
Division of Economics and Business
1500 Illinois Street
Golden, CO, 80401
E-mail: srebenna@mines.edu

constraints · non-overlap constraints · design problem · polylithic solution approach · computational geometry

1 Introduction

In an extension of the work by Kallrath (2009, [11]), we cut a set of ellipses with given semi-major and semi-minor axes from a rectangular plate. The ellipses are to be placed, free of any orientation restrictions, on a rectangular plate such that the area of the rectangle is minimized. The ellipses are not allowed to overlap; which poses the major challenge of this cutting problem together with the free rotation of the ellipses. Minimizing the area of the design rectangle is equivalent to minimizing trimloss.

Part of the motivation of this work is pure mathematical curiosity – ellipse cutting problems have a variety of real world applications. Color to be painted on walls is usually stored in paint buckets, which have an elliptical shape. This shape allows the painting rolls to be larger in length compared to a circular bucket having the same volume. The best area utilization is desired when transporting such buckets (differently or equally sized) on palettes. As described by Miller (2012, [20]), ellipses can be used to approximate (irregular or non-convex) geometric objects via a cover. The obtained ellipse placements can then help to compute solutions to the original problem and provide safe bounds on optimal solutions, *e.g.*, on the area of the design rectangle to be minimized.

The ellipse cutting problem falls into the class of two-dimensional cutting or packing problems of regular objects. This cutting problem comes close to the 2/V/D/F classification of Dyckhoff (1990, [4]); *i.e.*, two-dimensional, V = a kind of assignment: a selection of objects and all items, D = an assortment of large objects: different figures, and F = an assortment of small items: few items of different figures. Packing and cutting problems differ in the following two aspects: (1) In cutting problems one tries to minimize trimloss or area, while in packing an area is given and one wants to fit as many objects as possible, and (2) while free objects are allowed in cutting problems, this might lead to stability problems in packing problems. Our ellipse problem falls into the category of a cutting problem.

The separation of ellipses by hyperplanes has been treated in a senior thesis by Miller (2012, [20]). His formulation starts from elementary geometry of ellipses from which he derives the hyperplane conditions. Confinement of the ellipses to the rectangle is modeled as a special hyperplane representing the boundaries of the rectangle. Our approach starts from a generic vector representation (to allow for its extension to the 3-D case in future work). At first, we compute the minimal and maximal extension of shifted and rotated ellipses. Second, we derive the hyperplane conditions from rotated coordinate systems. Miller reports in his thesis only on small examples with two ellipses. This is confirmed by his advisor Floudas (2012, [6]).

Other than the unpublished work by Miller, it is hard to find published work on numerical approaches towards ellipse cutting and packing. However, the related problems of circle packing/cutting or orientation-free rectangle packing/cutting into one or multiple (design) rectangles have a rich body of literature, see Kallrath (2009, [11]) and Rebennack et al. (2009, [24]) and the references therein.

The technical report by Gensane and Honvault (2012, [8]) establishes optimal packings for the case of two congruent ellipses in a square. Based on theoretical developments for sphere packings, the authors are able to derive the position for two congruent ellipses along with the minimal side length of a square hosting those two ellipses. Thus, their work is theoretical and not algorithmic, like ours.

The contributions of this paper are twofold: We develop

1. novel mathematical programming models for the ellipse cutting problems which allow us to solve larger instances to global optimality than previously reported in the literature.
2. two polylithic[1] approaches to compute good and near optimal cuttings for instances which cannot be handled by the current nonlinear and global solvers for the exact mathematical programming formulation developed. Both approaches sequentially solves ellipse cutting problems with fewer number of ellipses.

The remainder of this article is organized as follows. In Sect. 2, we develop (MI)NLP models for cutting ellipses from the design rectangle. We construct our polylithic approaches to compute good feasible, near optimal cuttings for instances which cannot be handled in the monolith formulation with the currently available solvers in Sect. 3. We present numerical experiments and results in Sect. 4. Sect. 5 concludes the paper. We summarize our notation in Appendix A.

At all places in this paper we use the term global optimum, or global optimality, in the sense of small relative gaps (difference between upper and lower bound divided by the lower bound) of the order of 10^{-5}. We are aware that the numeric solvers dealing with finite number arithmetic are subject to round-off errors. As such, all the presented results are only approximations, and although the small gaps hint on good/optimal results, due to the rounding errors, we do not obtain reliable results in the sense of interval analysis. For packing circles in a unit square, we find high precision guaranteed enclosures for both the global optimizer and the global optimum value and details of the interval arithmetic-based core elimination method in a series of publications by Markót and Csendes (2005,[19]) and Markót (2007,[18]).

[1] The expression *polylithic* has been introduced by Kallrath (2009, [10]; 2011, [12]) to refer to modeling and solution approaches in which optimization problems are solved by tailor-made methods involving several models or solve statements or algorithmic components.

2 Monolithic: Non-convex (MI)NLP Models

We describe ellipses by the coordinates of their center and an orientation angle to allow for their rotation. We need to model two types of constraints: (1) non-overlap of ellipses and (2) bounds placing the ellipses inside the rectangle.

While non-overlap of two circles can be enforced by one non-convex constraint, ensuring that their centers are apart no less than the sum of their radii, the case for ellipses is more involved (one reason is the possibility for rotation). We use the following key idea to ensure non-overlap: Because ellipses are convex objects, there must exist a hyperplane (*i.e.*, a line) between any two pairs of ellipses, separating them from each other.

When the context allows, then we utilize a vector notation using the Euclidean norm scalar products to avoid the additional dimension index d. The vector notation is indicated by bold symbols. We use lower case symbols for variables, and upper case symbols for input or derived data. The only exceptions are the semi-major and semi-minor axes a_i and b_i, respectively, of the ellipses and the model indices.

Note that we provide lower and upper bounds in the model wherever possible and as tight as possible as these bounds help to solve the NLP and MINLP problems to global optimality.

We start with the modeling of the non-overlap and boundary constraints for ellipses in Sect. 2.1. This leads us to two equivalent NLP models, as summarized in Sect. 2.2. To enhance computational efficiency, we then present symmetry breaking constraints (Sect. 2.3), mixed-integer extensions (Sect. 2.4), and lower/upper bounding problems (Sect. 2.5).

2.1 Towards the NLP Model Formulations

The objective function minimizes the area, a, of the design rectangle

$$\min\ a, \quad a = x_1^{\mathrm{R}} x_2^{\mathrm{R}} \quad , \tag{1}$$

where decision variable x_d^{R} represents the extension of the design rectangle in dimension d; x_1^{R} denotes the length and x_2^{R} is the with of the rectangle.

Equivalently to (1), we could minimize waste, *i.e.*,

$$\min\ z, \quad z = a - \sum_{i \in \mathcal{I}} A_i \quad , \tag{2}$$

where A_i denotes the area of ellipse i; set $\mathcal{I}$ is the collection of ellipses to be packed.

The extensions x_d^{R} of the rectangle are subject to pre-given bounds, S_d^- and S_d^+

$$S_d^- \le x_d^{\mathrm{R}} \le S_d^+ \quad , \quad \forall d \quad . \tag{3}$$

The upper bound, S_d^+, could be motivated by technical limitations; a lower bound, S_d^+, is given by the maximum of all the minor ellipse axis lengths (maximum of $2b_i$ over all i). Refinements of these bounds are described in Sect. 2.5, where we exploit circle cuttings.

2.1.1 Cutting Circles

We start with the modeling of the circle cutting problem for two reasons: first, it was the starting point for our analysis of the ellipses cutting problem, and second, we use it to compute valid lower and upper bounds on the ellipse cutting problem (*cf.* Sect. 2.5).

The non-overlap constraints for circles i and j read

$$\left\|\mathbf{x}_i^0 - \mathbf{x}_j^0\right\|_2^2 := \sum_{d=1}^{2} \left(x_{id}^0 - x_{jd}^0\right)^2 \geq (R_i + R_j)^2 \quad , \quad \forall\{ij : i < j\} \quad , \tag{4}$$

with radius R_i and (decision variable) x_{id}^0 modeling the center of circle i in dimension d. Constraints (4) are non-convex constraints (the left hand side constitutes a convex function). Note that for n circles we have $n(n-1)/2$ inequalities of type (4).

Fitting the circles inside the enclosing rectangles requires

$$x_{id}^0 \geq R_i \quad , \quad \forall\{id\} \tag{5}$$

and

$$x_{id}^0 + R_i \leq x_d^{\mathrm{R}} \quad , \quad \forall\{id\} \quad . \tag{6}$$

2.1.2 Cutting Ellipses

One might be tempted to follow the idea of the non-overlapping conditions (4) when treating ellipses. Unfortunately, the known radii R_i and R_j for the cases of circles become orientation-dependent variables. It turns out that this approach is not ideal, from the perspective of mathematical programming modeling. Thus, we follow a different idea.

Ellipse i is characterized by its semi-major and semi-minor axis a_i and b_i, respectively. The ellipses i will be implicitly described by their centers, and their orientations; see Fig. 1. The ellipses can be placed at a free "center" represented by the vector $\mathbf{x}_i^0$ (with components x_{id}^0) with the semi-major axis a_i inclined by the angle θ_i. For $\theta_i = 0$, the ellipse is characterized by the equation

$$\frac{(x_1 - x_{i1}^0)^2}{a_i^2} + \frac{(x_2 - x_{i2}^0)^2}{b_i^2} = 1 \quad , \tag{7}$$

i.e., all points $(x_1, x_2) \in \mathbb{R}^2$ satisfying constraint (7) lie on the perimeter of ellipse i. For the rotated ellipse i, we exploit the coordinate transformation

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \mathsf{R}_{\theta i} \begin{pmatrix} x_1 - x_{i1}^0 \\ x_2 - x_{i2}^0 \end{pmatrix}$$

with

$$\mathsf{R}_{\theta i} := \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \tag{8}$$

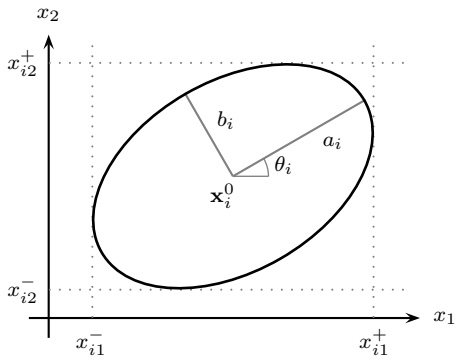


Fig. 1 Representation of ellipses i: center $\mathbf{x}_i^0$, semi-major axis a_i, semi-minor axis b_i, rotation angle θ_i and extrema x_{i1}^-, x_{i2}^-, x_{i1}^+, and x_{i2}^+.

or equivalently

$$\begin{aligned} x_1' &= (\cos\theta_i)(x_1 - x_{i1}^0) - (\sin\theta_i)(x_2 - x_{i2}^0) \\ x_2' &= (\sin\theta_i)(x_1 - x_{i1}^0) + (\cos\theta_i)(x_2 - x_{i2}^0) \quad . \end{aligned}$$

Now we can insert x_1' and x_2' into the ellipse equation (7).

More generally, an arbitrarily oriented ellipse i, centered at $\mathbf{x}_i^0 \in \mathbb{R}^2$, is defined by the equation (quadratic form)

$$(\mathbf{x} - \mathbf{x}_i^0)^\top \mathsf{A}_i (\mathbf{x} - \mathbf{x}_i^0) = 1 \quad , \tag{9}$$

where A_i is a positive definite matrix and $\mathbf{x} \in \mathbb{R}^2$.

The eigenvectors of A_i define the principal directions of the ellipse (or, ellipsoid in 3-D) and the eigenvalues of A_i are the inverse squares of the semi-axes: a_i^{-2} and b_i^{-2}. An invertible linear transformation applied to a circle (sphere) produces an ellipse (ellipsoid), which can be brought into the above standard form by a suitable rotation, a consequence of the polar decomposition (*cf.* Spectral Theorem). If the linear transformation is represented by a symmetric 2-by-2 (3-by-3) matrix, then the eigenvectors of the matrix are orthogonal (due to the Spectral Theorem) and represent the directions of the axes of the ellipse (ellipsoid): the lengths of the semi-axes are given by the eigenvalues. In detail, for ellipse i with semi-axes a_i and b_i rotated by $\mathsf{R}_{\theta i}$ as defined in (8), we have

$$\mathsf{A}_{\theta i} := \mathsf{R}_{\theta i} \mathsf{D}_i \mathsf{R}_{\theta i}^\top \tag{10}$$

with

$$\mathsf{D}_i := \begin{pmatrix} \lambda_{i1} & 0 \\ 0 & \lambda_{i2} \end{pmatrix} = \begin{pmatrix} a_i^{-2} & 0 \\ 0 & b_i^{-2} \end{pmatrix} \quad . \tag{11}$$

To avoid the occurrence of trigonometric terms in the optimization model, we use the following transformation into an equivalent (non-convex) quadratic model. We replace the decision variable θ_i by the two decision variables

$$v_i := \cos\theta_i \qquad \text{and} \qquad w_i := \sin\theta_i \quad .$$

The new variables are subject to the bounds

$$-1 \leq v_i \leq +1 \qquad \text{and} \qquad -1 \leq w_i \leq +1$$

and they are coupled by the Pythagorean Theorem

$$v_i^2 + w_i^2 = 1 \quad . \tag{12}$$

Note that for ellipses, due to their symmetry, it suffices to consider rotation angles, θ_i, in the range of $0°$ to $180°$, *i.e.*, $0 \leq w_i \leq 1$.

With this notation, we obtain

$$\begin{aligned} \mathsf{A}_{\theta i} &= \begin{pmatrix} v_i & -w_i \\ w_i & v_i \end{pmatrix} \begin{pmatrix} \lambda_{i1} & 0 \\ 0 & \lambda_{i2} \end{pmatrix} \begin{pmatrix} v_i & w_i \\ -w_i & v_i \end{pmatrix} \\ &= \begin{pmatrix} v_i^2\lambda_{i1} + w_i^2\lambda_{i2} & v_i w_i \lambda_{i1} - v_i w_i \lambda_{i2} \\ v_i w_i \lambda_{i1} - v_i w_i \lambda_{i2} & v_i^2\lambda_{i2} + w_i^2\lambda_{i1} \end{pmatrix} \quad . \end{aligned}$$

Fitting the ellipses inside the enclosing design rectangle requires that

$$0 \leq x_{id}^- \leq x_{id}^+ \leq x_d^{\mathrm{R}} \quad , \quad \forall\{id\} \quad , \tag{13}$$

where x_{id}^- and x_{id}^+ are the minimum and maximum extensions of ellipse i in dimension d.

In Sect. 2.1.3, we show that

$$\begin{aligned} x_{i1}^{\pm} &= x_{i1}^0 \pm \sqrt{a_i^2 \cos^2\theta_i + b_i^2 \sin^2\theta_i} \quad \text{and} \\ x_{i2}^{\pm} &= x_{i2}^0 \pm \sqrt{b_i^2 \cos^2\theta_i + a_i^2 \sin^2\theta_i} \quad . \end{aligned}$$

For instance, $\theta_i = 0$ implies $x_{i1}^{\pm} = x_{i1}^0 \pm a_i$ and $x_{i2}^{\pm} = x_{i2}^0 \pm b_i$. Similarly, if the ellipses are circles ($a_i = b_i = r_i$), then for any angle θ_i we obtain $x_{i1}^{\pm} = x_{i2}^{\pm} = x_{i1}^0 \pm r_i$.

We continue with the derivation of these quantities in the following section; if you believe in the formulae above, then you can skip this section. The non-overlap conditions (discussed in Sect. 2.1.4) are then based on the ideas of extremal extensions of the ellipses.

2.1.3 Minimum and Maximum Extensions of Ellipses

Let us compute x_{id}^- and x_{id}^+ for ellipse i with center x_{id}^0 by solving the following optimization problems

$$x_{id}^- = \min\ \mathbf{c}^\top \mathbf{x} = \min\ x_{id} \quad , \quad \forall d$$

and

$$x_{id}^+ = \max\ \mathbf{c}^\top \mathbf{x} = \max\ x_{id} \quad , \quad \forall d \quad ,$$

respectively, subject to the ellipse condition (9); for $d = 1$ we select $\mathbf{c}^\top := (1, 0)$, while for $d = 2$ we have $\mathbf{c}^\top := (0, 1)$. Instead of using (9), we can solve the simpler optimization problem

$$x_{id}^- = \min\ \mathbf{c}^\top(\mathbf{x} + \mathbf{x}_i^0) = x_{id}^0 + \min\ x_{id} \quad , \qquad \forall d \tag{14}$$

and

$$x_{id}^+ = \max\ \mathbf{c}^\top(\mathbf{x} + \mathbf{x}_i^0) = x_{id}^0 + \max\ x_{id} \quad , \quad \forall d \quad , \tag{15}$$

respectively, subject to

$$\mathbf{x}^\top \mathsf{A}_{\theta i} \mathbf{x} = 1 \quad , \tag{16}$$

which describes an ellipse centered at the origin. Note, however, that ellipse i cannot be centered at the origin, as the origin is identical to the left-bottom corner of the design rectangle.

The Lagrangian function of both optimization problems (14) and (15) reads

$$\mathcal{L}(\mathbf{x}, \bar{\lambda}) = \mathbf{c}^\top(\mathbf{x} + \mathbf{x}_i^0) + \bar{\lambda}\left(\mathbf{x}^\top \mathsf{A}_{\theta i} \mathbf{x} - 1\right) \tag{17}$$

for an (unrestricted) Lagrangian multiplier $\bar{\lambda} \in \mathbb{R}$. The first order Karush-Kuhn-Tucker (KKT) conditions are derived as

$$\mathbf{c} + 2\bar{\lambda}\mathsf{A}_{\theta i}^\top \mathbf{x} = \mathbf{0} \tag{18}$$

together with (16). We multiply (18) by $\mathbf{x}^\top$ from the left side, (this operation is safe, as the center of ellipse i cannot be an extremum) and exploit (16) to obtain

$$x_d + 2\bar{\lambda} = 0 \quad , \quad \forall d \quad .$$

This allows us to eliminate the Lagrangian multiplier $\bar{\lambda}$ from (18) yielding

$$\mathbf{c} - x_d \mathsf{A}_{\theta i}^\top \mathbf{x} = 0 \quad , \quad \forall d \quad . \tag{19}$$

For the first dimension ($d = 1$) the two equations in (19) read

$$\begin{aligned} 1 - x_1\left(A_{11}x_1 + A_{21}x_2\right) &= 0 \\ -x_1\left(A_{12}x_1 + A_{22}x_2\right) &= 0 \end{aligned}$$

with

$$\mathsf{A}_{\theta i} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad .$$

As $x_1 \neq 0$ (the center of the shifted ellipse cannot be a stationary point of these KKTs), we derive

$$x_2 = -\frac{A_{12}}{A_{22}} x_1 \tag{20}$$

from which we further derive

$$x_1^2 = \left(A_{11} - A_{21}\frac{A_{12}}{A_{22}}\right)^{-1} = \frac{A_{22}}{\lambda_{i1}\lambda_{i2}} = A_{22} a_i^2 b_i^2 \quad , \tag{21}$$

where we exploit the fact that

$$\det \mathsf{A}_{\theta i} = A_{11}A_{22} - A_{12}A_{21} = \lambda_{i1}\lambda_{i2} > 0$$

(*cf.* Eigenvector Decomposition). From the geometry of the optimization problems (14) and (15), we know that each problem has a unique, global extremum. We further know that the global extrema satisfy the KKT conditions (16) and (18) (*i.e.*, they are necessary). Because we have not excluded any global optima in our deviation to derive at (20) and (21) and they lead to exactly two points, we know that x_1 and x_2 in (20) and (21) define the global optimum for (15) and (14); one just needs to pick the correct one.

The minimum and maximum extensions of ellipse i in the first dimension, $(d = 1)$, then reduce to

$$\begin{aligned} x_{i1}^- &= \min\ \mathbf{c}^\top(\mathbf{x} + \mathbf{x}_i^0) \\ &= x_{i1}^0 - \sqrt{x_1^2} \\ &= x_{i1}^0 - a_i b_i \sqrt{A_{22}} \\ &= x_{i1}^0 - \sqrt{a_i^2 \cos^2\theta_i + b_i^2 \sin^2\theta_i} \end{aligned} \tag{22}$$

and

$$x_{i1}^+ = x_{i1}^0 + \sqrt{a_i^2 \cos^2\theta_i + b_i^2 \sin^2\theta_i} \quad , \tag{23}$$

respectively.

Similarly, for $d = 2$ we derive

$$\begin{aligned} -x_2\,(A_{11}x_1 + A_{21}x_2) &= 0 \\ 1 - x_2\,(A_{12}x_1 + A_{22}x_2) &= 0 \end{aligned}$$

to obtain

$$x_2^2 = \left(A_{22} - A_{12}\frac{A_{21}}{A_{11}}\right)^{-1} = \frac{A_{11}}{\lambda_{i1}\lambda_{i2}} = A_{11}a_i^2 b_i^2 \quad .$$

This leads to

$$x_{i2}^- = x_{i2}^0 - \sqrt{b_i^2 \cos^2\theta_i + a_i^2 \sin^2\theta_i} \tag{24}$$

and

$$x_{i2}^+ = x_{i2}^0 + \sqrt{b_i^2 \cos^2\theta_i + a_i^2 \sin^2\theta_i} \quad . \tag{25}$$

2.1.4 Non-Overlap Condition for Ellipses

One might have the following idea to model the non-overlap constraints for pairs of ellipses: We could enter a few points $\mathbf{x}_j$ on the circumference of ellipse j into the equation defining ellipse i. We would then ask that

$$(\mathbf{x}_j - \mathbf{x}_i^0)^\top \mathsf{A}_{\theta i} (\mathbf{x}_j - \mathbf{x}_i^0) \geq 1 \quad ; \tag{26}$$

the "$\geq$" forces the circumferences of ellipse j not to enter ellipse i. However, to ensure non-overlap of the two ellipses i and j in this way, we would need to ensure that the (non-convex) constraints (26) hold for a continuum of points (not just a few), leading to a semi-infinite programming problem. One could resort to similar ideas of Rebennack and Kallrath (2013, [22], [23]) or consult one of the various surveys about semi-infinite programming, for instance, by Hettich and Kortanek (1993, [9]) or Lopez and Still (2007, [16]). We take a different route.

In Sect. 2.1.3, we derive the formulae to compute x^{-}_{id}, the minimum value the ellipse extends to in coordinate axis direction d. This computation can be extended by the following idea. Assume we are given a rotated ellipse i whose semi-major axis a_i has an angle θ_i with the x-axis (the length of the rectangle). Furthermore, we have a separating line (or, hyperplane) parameterized by

$$\mathbf{G}(t) := \mathbf{g}^0 + \mathbf{g} \cdot t \quad , \quad t \in \mathbb{R} \tag{27}$$

with footpoint $\mathbf{g}^0$, and direction $\mathbf{g}$ normalized to $|\mathbf{g}| = 1$. The footpoint $\mathbf{g}^0$ is not uniquely defined, because it can lie anywhere on the separating hyperplane. As such, $\mathbf{g}^0$ has two degrees of freedom. We eliminate one degree of freedom by requesting that that the footpoint lies on the intersection of the hyperplane and the line segment between the centers $\mathbf{x}^0_i$ and $\mathbf{x}^0_j$ of ellipses i and j. Therefore, we introduce the non-negative variable λ, $0 \leq \lambda_{ij} \leq 1$, and represent the footpoint as the linear combination

$$\mathbf{g}^0 = \lambda_{ij}\mathbf{x}^0_i + (1 - \lambda_{ij})\mathbf{x}^0_j \tag{28}$$

of the ellipse centers $\mathbf{x}^0_i$ and $\mathbf{x}^0_i$. With three decision variables (two for $\mathbf{g}^0$ and one for λ_{ij}) and two constraints, we are left with one degree of freedom instead of two.

Can we provide a necessary condition for the ellipse being completely above $\mathbf{G}(t)$, or just touching it? We can: $\mathbf{G}(t)$ has an inclination angle ω as derived from a scalar product of the unit vector $(1, 0)^\top$ and $\mathbf{g}$ as

$$\cos\omega := (1, 0) \cdot \mathbf{g} = g_1 \tag{29}$$

with the x_1-axis, and intersects with the x_1-axis at x_h. Note that this intersection point only exists and necessary in our considerations, if $\mathbf{g}$ is not parallel to the x_1-axis. The special parallel case does not depend on the coordinate transformation described below and allows us to compute the distance of the ellipse to the hyperplane directly. Note that for the moment we consider a generic hyperplane $\mathbf{G}(t)$. Later, for separating two ellipses i and j, $\mathbf{G}(t)$, and also ω, will become dependent on i and j.

For ellipse i and hyperplane $\mathbf{G}(t)$, we resort again to a coordinate transformation: We transform the coordinate system in such a way that (1) $(x_h, 0)$ becomes the origin of the new coordinate system and (2) $\mathbf{G}(t)$ becomes identical to the new x-axis. If we then represent the ellipse in the new coordinate system (translation and rotation), we can apply the formulae of Sec. 2.1.3 to compute the minimum extension of ellipse i in dimension $d = 2$ in the new

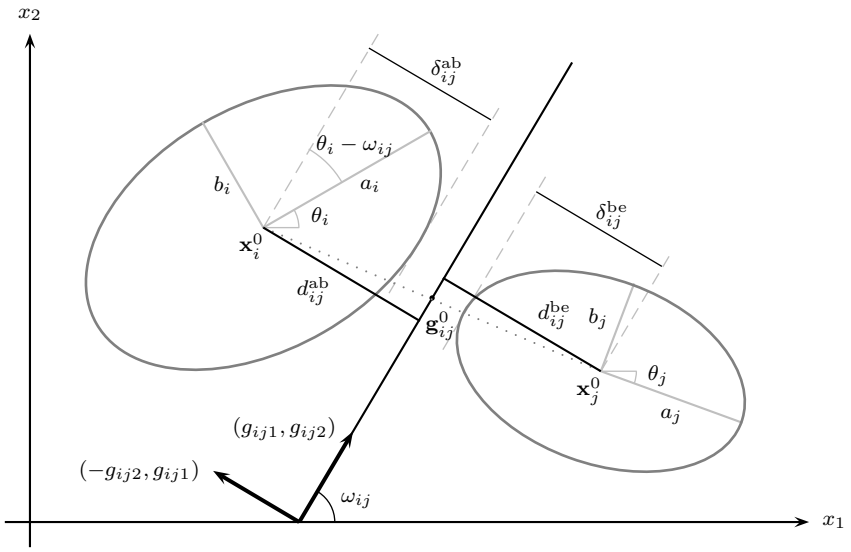


Fig. 2 Non-overlap of ellipses i and j via separating line.

Notation ellipses:

a_i, a_j semi-major axis	b_i, b_j semi-minor axis
θ_i, θ_j orientation angle	$\mathbf{x}_i^0, \mathbf{x}_j^0$ center

$\delta_{ij}^{\mathrm{ab}}$ maximal vertical extension of ellipse i to the hyperplane

$\delta_{ij}^{\mathrm{be}}$ maximal vertical extension of ellipse j to the hyperplane

Notation separating line:

$\mathbf{g}_{ij}^0$ footpoint	ω_{ij} inclination angle
(g_{ij1}, g_{ij2}) direction vector	$d_{ij}^{\mathrm{ab}}, d_{ij}^{\mathrm{be}}$ distance to line

coordinate system: $x_{i2}^{-\prime}$. Now, if $x_{i2}^{-\prime} \geq 0$, ellipse i lies above the hyperplane; $x_{i2}^{-\prime} \geq 0$ is also the necessary and sufficient condition for ellipse i to be above the hyperplane (or, just touching it).

Before we algebraically derive the coordinate transformation and the non-overlap conditions for each pair of ellipses, we built some intuition with Fig. 2. Using geometry, d_{ij}^{ab} is the length of the projection of the vector $(\mathbf{x}_i^0 - \mathbf{g}_{ij}^0)$ on the vector $(-g_{ij2}, g_{ij1})$. Since $(-g_{ij2}, g_{ij1})$ has unit length, the length of the projection is simply the inner product

$$d_{ij}^{ab} = -g_{ij2}(x_{i1}^0 - g_{ij1}^0) + g_{ij1}(x_{i2}^0 - g_{ij2}^0) \quad .$$

On the other hand, the length δ_{ij}^{ab} is the maximum vertical extension of ellipse i if we consider the hyperplane as the horizontal axis. Now – relative to the hyperplane – the angle of the ellipse is $\theta_i - \omega_{ij}$. So using equation (24), we obtain

$$\delta_{ij}^{ab} = \sqrt{b_i^2 \cos^2(\theta_i - \omega_{ij}) + a_i^2 \sin^2(\theta_i - \omega_{ij})} \quad .$$

For ellipse i to be "above" the hyperplane, we require $(d_{ij}^{ab})^2 \geq (\delta_{ij}^{ab})^2$.

Similarly, d_{ij}^{be} is the projection of the vector $(\mathbf{x}_j^0 - \mathbf{g}_{ij}^0)$ on the vector $(-g_{ij2}, g_{ij1})$; because $\mathbf{x}_j^0$ is on the opposite side of the hyperplane – in the minus $(-g_{ij2}, g_{ij1})$ direction – the inner product is negative. Moreover, from Fig. 2, it is clear that the absolute value of d_{ij}^{ab} and d_{ij}^{be} must be at least b_i.

Using the trigonometric identities for the cosine of the differences of angles, we can further derive

$$\begin{aligned}\cos(\theta_i - \omega_{ij}) &= \cos(\theta_i)\cos(\omega_{ij}) + \sin(\theta_i)\sin(\omega_{ij}) \\ &= g_{ij1}\cos(\theta_i) + g_{ij2}\sin(\theta_i) \quad ,\end{aligned}$$

which establishes the connection of the rotation angle θ_i of ellipse i and the inclination angle ω_{ij} of the hyperplane.

Now, let us derive the above equations algebraically. To establish the coordinate transformation for ellipse i and hyperplane $\mathbf{G}(t)$, let t_0 denote the value of t, for which we obtain $G_2(t) = 0$, *i.e.*, the separating line intersects with the original x-axis. In the new, shifted and rotated coordinate system the center of the ellipse i is given by

$$\mathbf{v}_i^0 := \begin{pmatrix} x_{i1}^0 - G_1(t^0) \\ d_i^0 \end{pmatrix} \quad ,$$

where d_i is the distance of the ellipse center to $\mathbf{G}(t)$, *i.e.*,

$$d_i := \frac{(-g_2, g_1)\cdot(\mathbf{x}_i^0 - \mathbf{g}^0)}{|\mathbf{g}|} = (-g_2, g_1)\cdot(\mathbf{x}_i^0 - \mathbf{g}^0) \quad , \tag{30}$$

where we constructed the vector $(-g_2, g_1)^\top$ orthogonal to $\mathbf{g}$. The ellipse, in the new coordinate system, can be generated by

$$\mathbf{v}_i(\varphi_i) = \mathbf{v}_i^0 + \mathsf{R}_{\theta-\omega,i}\begin{pmatrix} a_i\cos\varphi_i \\ b_i\sin\varphi_i \end{pmatrix} \quad , \quad 0 \le \varphi_i \le 2\pi \quad .$$

It fulfills the constraint

$$(\mathbf{v} - \mathbf{v}_i^0)^\top \mathsf{E}_{\theta-\omega,i}(\mathbf{v} - \mathbf{v}_i^0) = 1$$

with

$$\mathsf{E}_{\theta-\omega,i} := \mathsf{A}_{\theta-\omega,i} = \mathsf{R}_{\theta-\omega,i}\mathsf{D}_i\mathsf{R}_{\theta-\omega,i}^\top \quad .$$

The angle $\theta_i - \omega$ for ellipse i satisfies the relation

$$\cos(\theta_i - \omega) = (\cos\theta_i, \sin\theta_i)\cdot\mathbf{g} \quad , \tag{31}$$

and is the angle between the semi-major axis and the separating line.

With formula (24), we obtain

$$v_{i2}^- = v_{i2}^0 - \sqrt{v_2^2}$$

with

$$v_2^2 = \left(E_{22} - E_{12} \frac{E_{21}}{E_{11}} \right)^{-1} = \frac{E_{11}}{\lambda_{i1} \lambda_{i2}} = E_{11} a_i^2 b_i^2 \quad ,$$

or

$$v_{i2}^- = v_{i2}^0 - a_i b_i \sqrt{E_{11}} = v_{i2}^0 - a_i b_i \sqrt{\cos^2(\theta_i - \omega)\lambda_{i1} + \sin^2(\theta_i - \omega)\lambda_{i2}} \quad .$$

The transformation (translation and rotation) leads, eventually, to the final constraint that ellipse i is "above" the hyperplane

$$v_{i2}^0 - a_i b_i \sqrt{\cos^2(\theta_i - \omega)\lambda_{i1} + \sin^2(\theta_i - \omega)\lambda_{i2}} \ \geq 0 \quad ,$$

or equivalently

$$\left(v_{i2}^0\right)^2 - \left[b_i^2 \cos^2(\theta_i - \omega) + a_i^2 \sin^2(\theta_i - \omega)\right] \geq 0 \quad \wedge \quad v_{i2}^0 \geq 0 \quad . \tag{32}$$

Similarly,

$$v_{i2}^0 + a_i b_i \sqrt{\cos^2(\theta_i - \omega)\lambda_{i1} + \sin^2(\theta_i - \omega)\lambda_{i2}} \ \leq 0 \quad ,$$

or

$$\left(v_{i2}^0\right)^2 - \left[b_i^2 \cos^2(\theta_i - \omega) + a_i^2 \sin^2(\theta_i - \omega)\right] \geq 0 \quad \wedge \quad v_{i2}^0 \leq 0 \quad . \tag{33}$$

enforce ellipse i to stay "below" the separating line.

Finally, we are able to put everything together to state the non-overlap conditions for a pair of ellipses i and j. For each such pair, we require one separating line of type (27) and force ellipse i to stay above (recall, that touching is fine) that line via (32) and force ellipse j to stay below that line via (33). Recall that the decision variables (*i.e.*, d_i, ω, $\mathbf{g}$ and $\mathbf{g}^0$) related to the separating line carry now the double index ij. Especially, d_i becomes d_{ij}^{ab}, measuring the distance of ellipse i *above* the separating line, and $-d_{ij}^{\mathrm{be}}$, measuring the distance of ellipse j *below* the separating line. We generate only hyperplanes for $i < j$. This is illustrated in Fig. 2.

The case that ellipse j lies above ellipse i is covered by the reflected direction vector $-\mathbf{g}$. As we do not restrict $\mathbf{g}$, or its components, in sign, the direction is selected automatically when solving the problem. Thus, it is also automatically decided which ellipse lies above and which under the hyperplane.

The distances d_{ij}^{ab} and $-d_{ij}^{\mathrm{be}}$ of the centers of ellipse i and j to the hyperplane, indexed by ij, separating the ellipses i and j are bounded by

$$b_i \leq d_{ij}^{\mathrm{ab}} \leq \overline{D}_{ij} \quad , \quad -\overline{D}_{ij} \leq d_{ij}^{\mathrm{be}} \leq -b_i \quad , \quad \forall \{ij : i < j\} \tag{34}$$

as follows from the geometry, for instance,

$$\overline{D}_{ij} = \frac{\sqrt{(S_1^+)^2 + (S_2^+)^2} - b_i - b_j}{2} \quad , \quad \forall \{ij : i < j\} \quad .$$

The model is completed by

$$0 \leq v_{i2}^0 = d_{ij}^{\text{ab}} \qquad \text{and} \qquad 0 \geq v_{j2}^0 = d_{ij}^{\text{be}} \quad , \quad \forall\{ij : i < j\} \tag{35}$$

for each ellipse i and j, and the computation of ω_{ij} according to (29).

Let us conclude this section with a structural comment which illuminates the non-overlap constraints from a geometrical point of view and connects the non-overlap constraints to the non-convex character of the problem. The non-overlap constraints lead to a geometrical situation with a non-convex domain: Imagine the rectangle, from which to cut the n ellipses, and assume that ellipse i is fixed. Consider another ellipse $j \neq i$. Then the feasible area of the center coordinate of ellipse j is a subset of the rectangle without the region covered by ellipse i.

Similar to the case for circles, for n ellipses we have $n(n-1)/2$ inequalities of type (32) and (33), each. However, there are the additional constraints to model the separating lines. We analyze the number of variables and constraints involved in the non-overlap constraints for ellipses in the following section.

2.2 The NLP Formulations

Finally, we are able to state the resulting non-convex NLP formulations. However, before we start with the mathematical programming problem involving trigonometric terms, we define, for notational ease,

$$S_{id1} := \begin{cases} a_i^2, & d = 1 \\ b_i^2, & d = 2 \end{cases} \qquad \text{and} \qquad S_{id2} := \begin{cases} b_i^2, & d = 1 \\ a_i^2, & d = 2 \end{cases} \quad , \quad \forall\{id\} \quad . \tag{36}$$

The ellipse cutting problem (EP) is then summarized as follows

$$(\text{EP}_\theta): \quad a^* = \min \prod_{d=1}^{2} x_d^{\text{R}} \tag{37}$$

subject to

(fit ellipses i into rectangle)

$$S_{id1} \cos^2 \theta_i + S_{id2} \sin^2 \theta_i \leq \left(x_{id}^0\right)^2 \qquad \forall\{id\} \tag{38}$$

$$S_{id1} \cos^2 \theta_i + S_{id2} \sin^2 \theta_i \leq \left(x_d^R - x_{id}^0\right)^2 \qquad \forall\{id\} \tag{39}$$

$$x_{id}^0 \leq x_d^R - b_i \qquad \forall\{id\} \tag{40}$$

(non-overlap of ellipses i and j)

$$\sum_{d=1}^{2} \left(g_{ijd}\right)^2 = 1 \qquad \forall\{ij : i < j\} \quad (41)$$

$$g^0_{ijd} = \lambda_{ij} x^0_{id} + (1 - \lambda_{ij}) x^0_{jd} \qquad \forall\{ijd : i < j\} \quad (42)$$

$$d^{\mathrm{ab}}_{ij} = -g_{ij2}\left(x^0_{i1} - g^0_{ij1}\right) + g_{ij1}\left(x^0_{i2} - g^0_{ij2}\right) \qquad \forall\{ij : i < j\} \quad (43)$$

$$d^{\mathrm{be}}_{ij} = -g_{ij2}\left(x^0_{j1} - g^0_{ij1}\right) + g_{ij1}\left(x^0_{j2} - g^0_{ij2}\right) \qquad \forall\{ij : i < j\} \quad (44)$$

$$\cos(\theta_i - \omega_{ij}) = g_{ij1}\cos\theta_i + g_{ij2}\sin\theta_i \qquad \forall\{ij : i < j\} \quad (45)$$

$$\cos(\theta_j - \omega_{ij}) = g_{ij1}\cos\theta_j + g_{ij2}\sin\theta_j \qquad \forall\{ij : i < j\} \quad (46)$$

$$\left(d^{\mathrm{ab}}_{ij}\right)^2 \geq b_i^2 \cos^2(\theta_i - \omega_{ij}) + a_i^2 \sin^2(\theta_i - \omega_{ij}) \qquad \forall\{ij : i < j\} \quad (47)$$

$$\left(d^{\mathrm{be}}_{ij}\right)^2 \geq b_j^2 \cos^2(\theta_j - \omega_{ij}) + a_j^2 \sin^2(\theta_j - \omega_{ij}) \qquad \forall\{ij : i < j\} \quad (48)$$

(variable domain)

$$S^-_d \leq x^{\mathrm{R}}_d \leq S^+_d \qquad \forall d \quad (49)$$

$$b_i \leq x^0_{id} \leq S^+_d - b_i \qquad \forall\{id\} \quad (50)$$

$$0 \leq \theta_i \leq \pi \qquad \forall i \quad (51)$$

$$0 \leq \lambda_{ij} \leq 1 \qquad \forall\{ij : i < j\} \quad (52)$$

$$0 \leq g^0_{ijd} \leq S^+_d \qquad \forall\{ijd : i < j\} \quad (53)$$

$$-1 \leq g_{ijd} \leq 1 \qquad \forall\{ijd : i < j\} \quad (54)$$

$$0 \leq \omega_{ij} \leq 2\pi \qquad \forall\{ij : i < j\} \quad (55)$$

$$b_i \leq d^{ab}_{ij} \leq \overline{D}_{ij} \qquad \forall\{ij : i < j\} \quad (56)$$

$$-\overline{D}_{ij} \leq d^{\mathrm{be}}_{ij} \leq -b_i \qquad \forall\{ij : i < j\} \quad . \quad (57)$$

We start with the description of the decision variables involved in (EP_θ). We have two variables for the design rectangle: its length and with, x^{R}_d. Each ellipse is modeled by three decision variables: the two center coordinates, x^0_{id}, and the rotation angle, θ_i. For each pair of ellipses i and j with $i < j$, we have one hyperplane ij, modeled by five decision variables: the two footpoint coordinates, g^0_{ijd}, the linear combination variable λ_{ij} and the two dimensional slope, g_{ijd}. The coordinate transformation for ellipse i with respect to hyperplane ij requires the angles ω_{ij}. Finally, the distance of the two ellipses i and j, in the new coordinate transform, involve the two decision variables d^{ab}_{ij} and d^{be}_{ij}. Thus, for each pair of ellipses there are eight decision variables involved with the modeling of the non-overlap condition for ellipses.

We minimize with (37) the area of the design rectangle, *i.e.*, objective function (1).

The first group of constraints, (38)-(40), enforces that all ellipses stay inside the design rectangle. Constraints (38) ensure that the minimum extension of each ellipse is non-negative; using constraints (13), (22), and (24). Similarly,

constraints (39) and (40) ensure that the maximum extension of each ellipse is inside the rectangle; using constraints (13), (23), and (25).

The second group, (41)-(48), models the non-overlap of each pair of ellipses i and j. Constraints (41) normalize the slope of the hyperplane ij; constraints (42) place the footpoint of the hyperplane on the line segment of the two center coordinates of the corresponding ellipses; *i.e.*, (28). The distance of the two ellipses i and j to their corresponding hyperplane ij is computed via (43) and (44); *i.e.*, (30). The angle for the coordinate transformation with respect to the hyperplanes is given by (45) and (46); *i.e.*, (31). Finally, the non-overlap of the two ellipses is given via (47) and (48); *i.e.*, (32) and (33), respectively.

With the discussion above, for $n \geq 2$ ellipses, (EP_θ) involves $2 - \frac{1}{2}n + \frac{7}{2}n^2$ continuous decision variables and $\frac{3}{2}n + \frac{9}{2}n^2$ functional constraints, not counting the box constraints (49)-(57).

We formulated the ellipse cutting problem as the NLP problem (EP_θ), with the non-convex objective function (37) and the non-convex constraints (38), (39), (41), (43)-(48), leading to a non-convex feasible region.

In Sect. 2.1.3, we indicate how to transform (EP_θ) into an equivalent (and, thus, non-convex) quadratic model. Here it is:

$$
\begin{aligned}
(\text{EP}_{\text{QP}}): \quad a^* = \min \; & \prod_{d=1}^{2} x_d^{\text{R}} \\
\text{s.t.} \; & (40), (41) - (44) \\
& v_i^2 + w_i^2 = 1 && \forall i \\
& S_{id1} v_i^2 + S_{id2}\big(1 - v_i^2\big) \leq \big(x_{id}^0\big)^2 && \forall \{id\} \\
& S_{id1} v_i^2 + S_{id2}\big(1 - v_i^2\big) \leq \big(x_d^R - x_{id}^0\big)^2 && \forall \{id\} \\
& p_{ij}^{\text{ab}} = g_{ij1} v_i + g_{ij2} w_i && \forall \{ij : i < j\} \\
& p_{ij}^{\text{be}} = g_{ij1} v_j + g_{ij2} w_j && \forall \{ij : i < j\} \\
& \big(d_{ij}^{\text{ab}}\big)^2 \geq b_i^2 \big(p_{ij}^{\text{ab}}\big)^2 + a_i^2 \Big(1 - \big(p_{ij}^{\text{ab}}\big)^2\Big) && \forall \{ij : i < j\} \\
& \big(d_{ij}^{\text{be}}\big)^2 \leq b_j^2 \big(p_{ij}^{\text{be}}\big)^2 + a_j^2 \Big(1 - \big(p_{ij}^{\text{be}}\big)^2\Big) && \forall \{ij : i < j\} \\
& (49), (50), (53), (54), (56), (57) \\
& -1 \leq v_i \leq 1 && \forall i \\
& 0 \leq w_i \leq 1 && \forall i \\
& -1 \leq p_{ij}^{\text{ab}}, p_{ij}^{\text{be}} \leq 1 && \forall \{ij : i < j\}
\end{aligned}
$$

We have re-formulated (EP_θ) as a quadratic optimization problem in form of (EP_{QP}) in order to utilize algorithms, specializing in bilinear and quadratic terms, which might be superior to general purpose algorithms and software packages.

The review by Floudas *et al.* (2005, [7]) and the paper by Misener & Floudas (2012, [21]) are good resources for further references and a description of various approaches to solve problems of the type of (EP_{QP}); we avoid repeating

the material here but list a few of the relevant references among them Androulakis *et al.* (1995, [3]), Maranas & Floudas (1995, [17]), Adjiman *et al.* (1996, [2]), and Adjiman *et al.* (2000, [1]). Algebraic reformulations and convex relaxation techniques as described in Liberti (2004, [13]), and Liberti & Pantelides (2006, [14]) are part of the global mixed-integer quadratic optimizer GloMIQO by Misener & Floudas (2012, [21]).

In the remainder of the paper, we refer to the ellipse cutting problem as (EP) and mean either formulation (EP_θ) or ($\mathrm{EP}_{\mathrm{QP}}$); any discussion on (EP) applies for both formulations equally.

Next, we enhance (EP) by symmetry breaking constraints (Sect. 2.3) and extensions using binary decision variables (Sect. 2.4).

2.3 Symmetry Breaking

The occurrence of symmetry in (any) mathematical programming problem can pose major challenges for global solvers for closing the optimality gap (this is also true for MILP solvers). Often, two symmetric solutions are "physically" identical (*e.g.*, when cutting identical objects) or can be mapped to each other via a point or axis reflection. Thus, breaking symmetry does not exclude interesting optimal solutions but may help to solve the problem instance at hand faster (or even at all). In the following, we address three such symmetries for our ellipse cutting problem and how to break them.

Given any optimal (ellipse) cutting and the corresponding rectangle, we can obtain several alternative optimal solutions by horizontal and vertical reflections; the solver views them as different solutions. We break this symmetry by requesting that the center of *one* of the ellipses is placed into the first quadrant of the design rectangle. Let ι be the index of that ellipse. Then, the symmetry breaking inequalities read

$$x^0_{\iota d} \leq \frac{1}{2} x^{\mathrm{R}}_d \quad , \quad \forall d \quad .$$

If congruent (*i.e.*, identical) ellipses are to be packed, then we break the resulting symmetry by sorting their center points with respect to the *lower left corner* of the rectangle. We collect all pairs (i, j) of congruent ellipses in the set $\mathcal{I}^{\mathrm{co}}$ (we assume ordered pairs $i < j$) and apply the ordering inequalities

$$x^0_{i1} + 5x^0_{i2} \leq x^0_{j1} + 5x^0_{j2} \quad , \quad \forall (i, j) \in \mathcal{I}^{\mathrm{co}} \quad . \tag{58}$$

Constraints (58) can be strengthened via lexicographic sorting involving mixed-integer programming techniques (*cf.* Sect. 2.4).

Rather a matter of degeneracy than of symmetry are free ellipses, *i.e.*, ellipses which can be moved locally without changing the objective function value (the area of the rectangle). In a cutting problem, free objects cause mainly degeneracy, however, in a packing problem, this poses major difficulties as they can freely "flow" around. We avoid free ellipses by adding a soft penalty term which moves the center coordinate towards the lower left corner of the rectangle.

2.4 MINLP Extensions

The NLP models presented in Sect. 2.2 contain a combinatorial component: The ellipses can be placed in any order to each other. Actually, for n ellipses, there are $n!$ many such orderings. This combinatorial nature is one of the reasons why the ellipse cutting problem turns out to be so computationally challenging to solve. We enhance the visibility of this combinatorial structure to the solvers via the following extension of the NLP models developed so far.

We partition the rectangle into a uniform grid of small rectangles. The size of the small rectangles depends on the problem instance and is governed by the "smallest" ellipses. It is chosen such that the center of each ellipse i can be uniquely assigned to one of these small rectangles. We denote by (c_x, c_y) one such small rectangle ("cell") and collect all cells in the set $\mathcal{I}^{\text{ce}}$. We control the assignment of ellipses to cells by the binary variables $\delta_{ic_xc_y}$. The following set of constraints assigns each ellipse to exactly one cell

$$\sum_{(c_x,c_y)\in\mathcal{I}^{\text{ce}}} \delta_{ic_xc_y} = 1 \quad , \quad \forall i \quad . \tag{59}$$

The centers of the ellipses are then subject to the constraints

$$x^0_{id} \geq C^-_{c_xc_yd} - S^+_d(1-\delta_{ic_xc_y}) \quad , \quad \forall\{\, i,d,(c_x,c_y)\in\mathcal{I}^{\text{ce}}\,\} \quad , \tag{60}$$

$$x^0_{id} \leq C^+_{c_xc_yd} + S^+_d(1-\delta_{ic_xc_y}) \quad , \quad \forall\{\, i,d,(c_x,c_y)\in\mathcal{I}^{\text{ce}}\,\} \quad , \tag{61}$$

where $C^-_{c_xc_yd}$ and $C^+_{c_xc_yd}$ are the lower and upper boundary coordinate of cell (c_x, c_y), respectively. If $\delta_{ic_xc_y} = 0$, then both constraints (60) and (61) (for each dimension) are dominated by the boundary conditions (13). However, $\delta_{ic_xc_y} = 1$ forces the center of ellipse i not to be outside of cell (c_x, c_y).

The ellipse cutting problem is now formulated via a MINLP problem. On paper, this MINLP looks even more difficult to solve than the NLP formulations. We might be surprised in Sect. 4.

The MINLP framework, and the presence of binary variables $\delta_{ic_xc_y}$, allows us to develop an enhancement for the symmetry breaking constraints (58) for identical ellipses (*cf.* Sect. 2.3).

Let i and j be the indices of two identical ellipses with the indices ordered as $i < j$, *i.e.*, $(i,j)\in\mathcal{I}^{\text{co}}$. We use a lexicographic ordering in the following sense: j is placed in a cell right of ellipse i, or it is in the same column of cells and above ellipse i (or the same). This condition is modeled as

$$\begin{aligned}\delta_{ic_xc_y} \leq & \sum_{c'_x:\, C_{c'_xc_y1}>C_{c_xc_y1}} \; \sum_{c'_y:\,(c'_x,c'_y)\in\mathcal{I}^{\text{ce}}} \delta_{jc'_xc'_y} \\ & + \sum_{c'_y:\, C_{c_xc'_y2}>C_{c_xc_y2}} \delta_{jc_xc'_y} \quad , \quad \forall\{\,(i,j)\in\mathcal{I}^{\text{co}},(c_x,c_y)\in\mathcal{I}^{\text{ce}}\,\}\end{aligned} \tag{62}$$

The first term models the "right" condition and the second term the "same column but higher" property.

2.5 Deriving Lower and Upper Bounds via Circle Cuttings

(Tight) lower and upper bounds on the minimal area of the design rectangle translate directly to lower and upper bounds on the length and width of the design rectangle; because of (3). Good bounds are crucial for the performance of global solvers.

To compute a lower bound on the minimal area of the design rectangle, we replace all ellipses by their inner circles, *i.e.*, $R_i = b_i$. We then compute the area-minimizing rectangle hosting all these inner circles with the formulation described in Sect. 2.1.1. Solving the resulting circle cutting problem is computationally relative easy compared to the corresponding ellipsoid cutting problem (note that both problems are in fact NP-hard). The solution of the circle cutting problem provides only a tight bound on the minimal area, when the semi-minor axes are not significantly smaller than the semi-major axes of all ellipses to be packed.

We obtain an upper bound on the minimal area of the design rectangle by replacing all ellipses by their outer circles, *i.e.*, $R_i = a_i$. We then compute the optimal design rectangle by solving the resulting circle cutting problem. By placing the centers of the ellipses at the locations of the center of the corresponding circles in an optimal circle cutting yields in a feasible ellipse cutting.

Generally, we denote a lower bound (upper bound) on the minimal area of the design rectangle by A^- (A^+) and the lower bound (upper bound) obtained by the inner circle (outer circle) cutting by $A^{\mathrm{ci},-}$ ($A^{\mathrm{ci},+}$).

We can now refine the lower and upper bounds on the rectangle length, L, and width, W, based on the cuttings we obtained for the inner and outer circle approximations. It is $L \cdot W \leq A^{\mathrm{ci},+}$ and $L \leq A^{\mathrm{ci},+}/W \leq A^{\mathrm{ci},+}/S_2^-$. The minimum width, S_2^-, of the rectangle, could be the maximum of all the minor ellipse axis lengths, yielding an upper bound on the rectangles length. Similarly, by using $A^{\mathrm{ci},-}$, we obtain a lower bound on the rectangles length.

3 Polylithic: Constructive Heuristics

If the number of ellipses increases, it becomes more and more difficult to compute a feasible point. Therefore, we have developed two polylithic approaches.

Both heuristics share the same idea: We sequentially add ellipses in a strip-packing fashion to the rectangle plate; we restrict the width of the rectangle, but leave its length unrestricted. We start with the placement of n_1 ellipses in the initial phase. Next, we use a greedy idea and add n_2 ellipses. In each step, we minimize the rectangle's area. The sequence, $s \in \mathcal{S}$, in which we add the ellipses is random. This adds some GRASP flavor to our approach (*cf.* Feo and Resende, 1995, [5]).

Some aspects of our heuristic may look similar to the *tiling method* introduced by Markót and Csendes (2005, [19]), but these similarities are accidental – one of the referees pointed us to this method during the review process.

The pseudo-code of the first algorithm, H1, looks as follows:

Input: Random sequence of ellipses, $\mathcal{S}$, parameters n_1 and n_2, time limits for solver (may be different for steps H1_1.1, H1_1.2.4, H1_1.2.5)

Output: Set of ellipse cuttings, lower bounds, and gaps

H1_1: For all sequences of ellipses $s \in \mathcal{S}$, do

H1_1.1: Solve (EP) for the first n_1 ellipses from s (usually, not solved to global optimality)

H1_1.2: While there are ellipses in the sequence s which have not been assigned, do

H1_1.2.1: For the best solution found (in steps H1_1.1 or H1_1.2.4 and H1_1.2.5) store obtained length of rectangle as $\bar{x}_1^{\mathrm{R}}$; fix the centers, $\mathbf{x}_i^0$, as well as the angles, θ_i, for all ellipses considered so far

H1_1.2.2: Choose the next (up to) n_2 ellipses – indexed by ι – in the sequence s

H1_1.2.3: Sequentially initialize the center(s), $\mathbf{x}_\iota^0$, and angle(s), θ_ι, of the new ellipse(s) according to the following formulae: $x_{\iota 1}^0 = \bar{x}_1^{\mathrm{R}} + b_\iota$, $x_{\iota 2}^0 = a_\iota$, and $\theta_\iota = 0$, *i.e.*, the additional ellipse(s) are added to the right of the existing ones; update $\bar{x}_1^{\mathrm{R}} \leftarrow \bar{x}_1^{\mathrm{R}} + 2b_\iota$; repeat for all n_2 ellipses ι

H1_1.2.4: Solve the resulting (EP)

H1_1.2.5: Unfix all center coordinates and angles; solve (EP) with the global solver selected; theoretically, that should exploit the current feasible point and improve on it

H1_1.3: Store feasible point (ellipse cutting) and lower bound computed in the latest solve of step H1_1.2.5; compute and store optimality gap

H1_1.4: Clear the model by removing all ellipses and their corresponding decision variables from (EP)

H1_2: Return set of feasible points, (safe) lower bounds and gaps.

The time limits for the (MI)NLP solves in steps H1_1.1, H1_1.2.4, and H1_1.2.5 make the algorithm a heuristic; typically the optimization problems are not solved to (proven) global optimality before the time limit is reached. We need to be careful to allow for enough CPU time (especially for steps H1_1.1 and H1_1.2.4) in order to find a feasible point for the given number of ellipses. If the ellipse cutting problem in step H1_1.2.5 can be solved to global optimality when all ellipses are present, then the ellipse cutting problem has been solved to global optimality as well. In most cases, however, the problem is not solved to global optimality in step H1_1.2.5 – after all, this is why we

use a polylithic approach. Rather, the idea is that the global solver is now getting the benefit of a good initial point and, like most solvers, it might start by doing local search from that point to improve upon the current point.

The idea of algorithm H1 is to aid the global solver by providing initial values for many of the decision variables, however, a few decision variables (*e.g.*, related to the separating lines) need to be determined by the solver. It turns out that, as the problems grow in size, feasible point(s) cannot be computed anymore – we have developed a second approach. H2 reads in pseudo-code as follows:

Input: Random sequence of ellipses, $\mathcal{S}$, parameters n_1, n_2, and n_3, time limits for solver (may be different for steps H2_1.2 and H2_1.2.4)

Output: Set of ellipse cuttings

H2_1: For all sequences of ellipses $s \in \mathcal{S}$, do

H2_1.1: Solve (EP) for the first n_1 ellipses from s (usually, not solved to global optimality)

H2_1.2: While there are ellipses in the sequence s which have not been assigned, do

H2_1.2.1: In the solution computed (in steps H2_1.2 or H2_1.2.4), identify the n_3 most right ellipses; we index them by r; fix their centers, $\mathbf{x}_r^0$, and $\cos(\theta_r)$; save the solution ($\mathbf{x}_i^0$, θ_i, g_{ij1}) for all ellipses considered in the last (EP); eliminate all ellipses already placed except for the n_3 most right ellipses (*i.e.*, the ellipses with index r remain in (EP))

H2_1.2.2: Choose the next (up to) n_2 ellipses – indexed by ι – in the sequence s

H2_1.2.3: The centers $\mathbf{x}_\iota^0$ of the n_2 ellipses are subject to the lower bound $X^- := \min\ x_{r1}^0$ (the smallest center coordinate among the n_3 ellipses in coordinate direction $d = 1$)

H2_1.2.4: Solve the resulting (EP), for the $n_2 + n_3$ ellipses

H2_1.3: Store feasible point (ellipse cutting) by restoring all previously computed values $\mathbf{x}_i^0$, θ_i, and g_{ij1}

H2_1.4: Clear the model by removing all ellipses and their corresponding decision variables from (EP)

H2_2: Return set of feasible points.

Heuristics H1 and H2 differ mainly in steps 1.2.1 and 1.2.3; step H1_1.2.5 is entirely missing in H2. The idea (and advantage in the computational speed compared to H1) of H2 is to sequentially solve (EP) of the same size; they

contain $n_2 + n_3$ ellipses. However, a critical tuning parameter is X^- used to prevent new ellipses from being placed left of the front. Note that H2 does not deliver a lower bound on the ellipse cutting problem, because the ellipses fixed in step H2_1.2.1 never get unfixed (within the same sequence s) but are not further considered when a new front is determined.

4 Numerical Experiments

We solve our non-convex MINLPs with the following global solvers available in GAMS: BARON [25], LindoGlobal [15] and GloMIQO [21]. All the instances used for the computations are summarized in Table 1. We used the following platforms for the computations.

Platform 1: Dual-six core machine with CPUs @ 2.93 GHz, 48GB RAM and 1TB HDD running Ubuntu 10.04.4.

Platform 2: Dual core machine with CPUs @ 2.5 GHz (Intel booth technology) 48GB RAM and 250 GB HDD running Windows 7.

Platform 3: Dual-six core machine with CPUs @ 3.3 GHz, 48GB RAM and 1TB HDD running Win2008 Server.

All computations utilize only a single core of the platforms specified above.

4.1 Proof-of-Concept: Treating Circles as Ellipses

We use the ellipse cutting formulation (EP_{QP}) to demonstrate the correctness of the approach by solving (published) circle cutting instances; circles are special cases of ellipses. We use test instances by Kallrath (2009, [11]). Note that there are some typos regarding the instances 5a and 5b as reported in Table 3 in [11]; we provide the correct radii of the circles here.

Consider now Table 2. In the first two columns, we report on the problem instance; a^* is the globally minimal area of the design rectangle as computed via a circle cutting formulation, *cf.* Sect. 2.1.1. In the other columns, we summarize the lower bound, A^-, and the upper bound, A^+, on the minimal area of the design rectangle as well as the computational time for (EP_{QP}) with three different global solvers available in GAMS. For these computations, for obvious reasons, we do not use the lower/upper bounds derived by the inner/outer circle cutting problems as described in Sect. 2.5. We observe that (1) the results computed with (EP_{QP}) are consistent with the global optima computed with the circle cutting formulation, (2) global optimality can only be proven for the instances with 5 circles within the time limit, (3) the globally optimal cuttings are only found for the cases of 5 circles, and (4) the global solvers perform very differently in terms of lower bounds and the quality of computed feasible solutions.

Two optimal circle cuttings are plotted in Fig. 3.

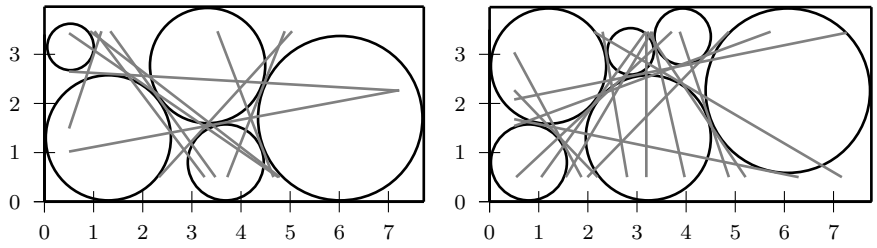
Table 1 Ellipse cutting instances.

test case	(a_i, b_i)	S_d^-	S_d^+	$\sum_i A_i$
Ellipse cutting instances "regular":				
TC02a	$(2, 1.5)$, $(1.5, 1)$	(0,0)	(8,4)	14.13717
TC02b	$(2, 1.5)$, $(1.8, 1.4)$	(0,0)	(8,4)	17.34159
TC03a	"TC02a" + $(1, 0.8)$	(0,0)	(8,4)	16.65044
TC03b	"TC02b" + $(0.8, 0.7)$	(0,0)	(8,4)	19.10088
TC04a	"TC03a" + $(0.9, 0.75)$	(0,0)	(8,4)	18.77102
TC04b	"TC03b" + $(1.1, 1)$	(0,0)	(14,4)	22.55664
TC05a	"TC04a" + $(0.8, 0.6)$	(0,0)	(8,4)	20.27898
TC05b	"TC04b" + $(0.9, 0.8)$	(0,0)	(14,4)	24.81858
TC06	"TC05a" + $(0.7, 0.3)$	(0,0)	(10,5)	20.93872
TC11	$(2, 1.5)$, $(1.8, 1.5)$, $(1.6, 1.5)$, $(1.5, 1.2)$, $(1.3, 1.0)$, $(1.2, 0.9)$, $(1.1, 0.8)$, $(1, 0.75)$, $(0.9, 0.6)$, $(0.8, 0.5)$, $(0.7, 0.3)$	(0,0)	(15,5)	47.31239
TC14	$7 \times (1, 0.75)$, $7 \times (0.5, 0.375)$	(0,0)	(8,4)	20.61670
TC20	"TC06" + $14 \times (1, 0.8)$	(0,0)	(30,5)	56.12455
TC30	"TC06" + $24 \times (1, 0.8)$	(0,0)	(30,5)	81.25729
TC50	"TC06" + $44 \times (1, 0.8)$	(0,0)	(60,5)	131.52278
TC100	"TC06" + $94 \times (1, 0.8)$	(0,0)	(120,5)	257.18648
Identical ellipses; total area of these n identical ellipses is $2\pi n$:				
TS02	$2 \times (2, 1)$	(0,0)	(8,4)	12.56637
TS03	$3 \times (2, 1)$	(0,0)	(8,4)	18.84956
TS04	$4 \times (2, 1)$	(0,0)	(8,4)	25.13274
TS05	$5 \times (2, 1)$	(0,0)	(12,5)	31.41593
TS06	$6 \times (2, 1)$	(0,0)	(12,5)	37.69911
TS07	$7 \times (2, 1)$	(0,0)	(15,5)	43.98230
TS08	$8 \times (2, 1)$	(0,0)	(20,5)	50.26548
TS09	$9 \times (2, 1)$	(0,0)	(20,5)	56.54867
TS10	$10 \times (2, 1)$	(0,0)	(20,5)	62.83185
TS11	$11 \times (2, 1)$	(0,0)	(22,5)	69.11504
TS12	$12 \times (2, 1)$	(0,0)	(24,5)	75.39822
TS13	$13 \times (2, 1)$	(0,0)	(26,5)	81.68141
TS14	$14 \times (2, 1)$	(0,0)	(28,5)	87.96459
TS15	$15 \times (2, 1)$	(0,0)	(30,5)	94.24778
Different eccentricities; $\rho \in [0, 1]$:				
TEρ	$(1.7, \rho 1.7)$, $(1.2, \rho 1.2)$, $(0.8, \rho 0.8)$	(0,0)	(8,4)	ρ15.61372
Circle cutting; [11, Table 3]:				
5a	(1.7,1.7), (1.3,1.3), (1.2,1.2), (0.5,0.5), (0.8,0.8)	(0,0)	(8,5)	21.70841
5b	"5a"	(0,0)	(18,5)	21.70841
6	"5a" + (0.6,0.6)	(0,0)	(8,4)	22.83948
7	"6" + (0.6,0.6)	(0,0)	(8,4)	23.97035
8	"5a" + (2.0,2.0), (1.3,1.3), (0.6,0.6)	(0,0)	(18,4)	40.71504
9	"8" + (0.6,0.6)	(0,0)	(18,4)	41.84601
10	"9" + (0.7,0.7)	(0,0)	(15,4)	43.38539

Table 2 Cutting circles with (EP_{QP}). We fix the rotation angles $\theta_i \equiv 0$. Test cases are taken from Table 3 in [11]. CPU time limit is 12 hours; `GAMS` 24.0.2; platform 1.

test case	a^*	**BARON** A^-	A^+	**h:mm**	**LindoGlobal** A^-	A^+	**h:mm**	**GloMIQO** A^-	A^+	**h:mm**
5a	30.62727	26.8025	–	†	***	***	1:46	30.62697	30.68817	†
5b	28.41130	28.4110	29.01239	†	***	***	8:40	***	***	0:07
6	30.62727	26.9677	–	†	30.62727	30.70490	†	22.83938	31.10945	†
7	31.12348	23.9704	–	†	23.97035	31.42927	†	23.97035	–	†
8	55.19630	40.7150	–	†	40.71504	56.53235	†	40.71504	58.14951	†
9	55.19630	41.8460	–	†	41.84601	57.14757	†	41.84601	63.44832	†
10	55.32089	43.3854	–	†	‡	‡	‡	43.38539	–	†

*** solved to proven global optimality (within 10^{-5})
† time limit reached
‡ solver error
– no feasible solution found



(a) Test case 5a (b) Test case 6

Fig. 3 Feasible circle cuttings computed via (EP_{QP}), including the separating lines.

4.2 Monolith

Table 3 summarizes the computational results for the monolith formulation (EP_{QP}). We observe that (1) all three current state-of-the-art global optimization solvers have difficulties closing the gap for the tested instances with three or more ellipses (with the given computational framework) and (2) good cuttings are computed, *cf.* Fig. 4.

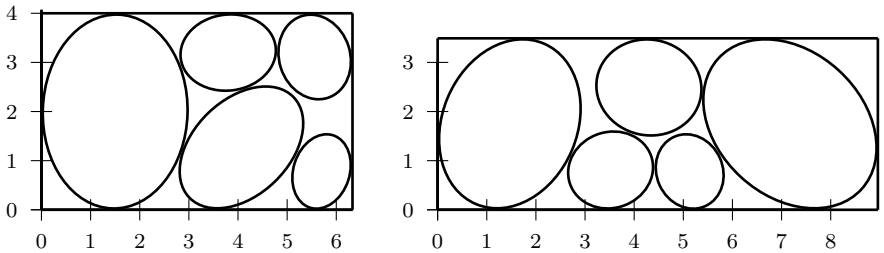
For case TC03, the smallest relative gaps, Δ, are obtained on platform 3 using LindoGlobal with `GAMS` 24.0.2 ($A^+ = 21.38577, A^- = 21.17972, \Delta = 0.00973$, 30 hours) and using GloMIQO with `GAMS` 23.8.2 ($A^+ = 21.38577, A^- = 20.58535, \Delta = 0.0388$, 63 hours).

Fig. 4(a) illustrates the difference between a packing and a cutting. The top right ellipse is not touching the ellipse centered at $(4.07, 1.27)$, which is allowed for cuttings but not for packings.

Table 3 Monolith: Cutting ellipses with ($\mathrm{EP}_{\mathrm{QP}}$). CPU time limit was 12 hours; `GAMS` 24.0.2; platform 1.

test case	BARON A^-	BARON A^+	BARON h:mm	LindoGlobal A^-	LindoGlobal A^+	LindoGlobal h:mm	GloMIQO A^-	GloMIQO A^+	GloMIQO h:mm
TC02a	***	18.00000	0:02	***	18.00000	0:01	***	18.00000	0:04
TC02b	***	22.23152	0:10	***	22.23152	0:08	***	22.23152	0:06
TC03a	17.04840	21.38576	†	17.32575	21.38577	†	17.04840	21.38577	†
TC03b	20.54006	25.22467	†	22.17848	25.22467	†	22.73854	25.22467	†
TC04a	19.26952	23.32845	†	19.26971	23.18708	†	19.26952	23.18774	†
TC04b	25.02665	29.22110	†	25.02690	28.54159	†	25.02665	28.54074	†
TC05a	20.27898	28.82368	†	20.27898	25.29557	†	20.27878	25.50112	†
TC05b	26.74996	33.84456	†	26.75023	31.28873	†	26.74996	31.28873	†
TC06	20.9387	–	†	20.93872	25.59380	†	20.93851	25.51043	†
TC11	47.3124	–	†	47.31238	64.59177	†	47.31239	74.95189	†
TC14	20.6165	–	†	20.61670	–	†	20.61650	29.65886	†

***	solved to proven global optimality (within 10^{-5})
†	time limit reached
–	no feasible solution found



(a) TC05a: $A^+ = 25.29557$ (b) TC5b: $A^+ = 31.28873$

Fig. 4 Ellipse cuttings computed via ($\mathrm{EP}_{\mathrm{QP}}$).

4.2.1 Inner and Outer Circles

Table 4 shows the lower bounds obtained by the area of ellipses ($\sum_i A_i$), the lower bounds obtained by the cutting computed for the inner circles ($A^{\mathrm{ci},-}$) and an upper bound derived from a cutting using outer circles ($A^{\mathrm{ci},+}$). The lower bounds obtained from the inner circle cuttings are slightly better than the sum of all areas of the ellipses when the majority of the ellipses has semi-minor axes not significantly smaller than their semi-major axes; otherwise, the sum of all areas of the ellipses exceeds $A^{\mathrm{ci},-}$. As the initial lower bounds obtained by the solvers for (EP) are significantly smaller, we use $\max\{\sum_i A_i, A^{\mathrm{ci},-}\}$ as the lower bound for (EP).

4.2.2 Tracking the Gap as a Function of the Eccentricity Ellipses

The more the ellipses deviate from circles, the more difficult it becomes to close the gap when solving (EP) to global optimality. To demonstrate this, we

Table 4 Comparing area of ellipses, inner circles, outer circles and best solution found (taken from Table 3 or Table 9).

test case	$\sum_i A_i$	$A^{\mathrm{ci},-}$	$A^{\mathrm{ci},+}$	A^{+}
TC02a	14.13716	14.84832	27.85641	18.00000$^{\flat}$
TC02b	17.34159	17.39465	30.37893	22.23152$^{\flat}$
TC03a	16.65044	17.04857	33.31371	21.38577
TC03b	19.10088	20.54006	34.08285	25.22467
TC04a	18.77102	19.26952	33.80990	23.18708
TC04b	22.55664	25.02665	37.79904	28.54159
TC05a	20.27898	19.82259	37.21122	25.29557
TC05b	24.81858	26.74996	41.48491	31.28873
TC06	20.93872	19.62543	37.36912	25.27463
TC11	47.31239	37.39281	83.04046	57.24034
TC14	20.61670	15.46253	35.48528	24.67185
$\flat$	proven global optimality (within 10^{-5})			

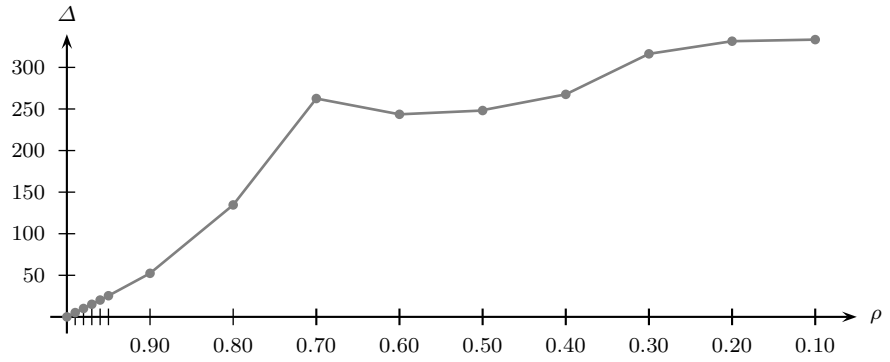


Fig. 5 Gap for TEρ instances for GloMIQO; see Table 5.

consider three ellipses with semi-major axes $a_1 = 1.7$, $a_2 = 1.2$, and $a_3 = 0.8$ and semi-minor axes $b_i = \rho a_i$ with $0 \le \rho \le 1$, these are the test cases TEρ of Table 1. The numerical eccentricity ε, as a measure for the deviation of the ellipses from circles, is connected to ρ by

$$\varepsilon = \frac{\sqrt{a^2 - b^2}}{a} = \sqrt{1 - \rho^2} \in [0, 1] \quad .$$

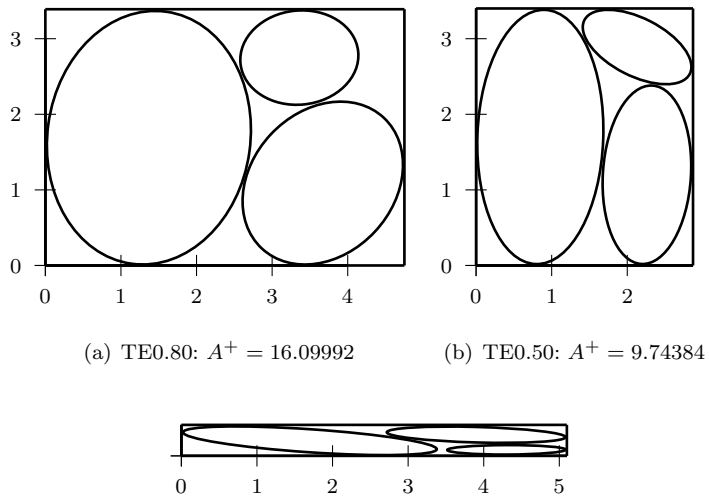
Computational results for some values of ρ are summarized in Table 5 displaying the computed lower bound, A^-, best solution found, A^+, and relative gap $\Delta = (A^+ - A^-)/A^-$ versus ρ for the three global solvers BARON, LindoGlobal, GloMIQO. The last column computes the fraction of area of the design rectangle covered by the ellipses, which is the utilization of the cutting.

Consider now Table 5. Even for a mild eccentricity of $\varepsilon = 0.199$, or $\rho = 0.98$, respectively, the gap is already very difficult to close. For $\rho \ge 0.7$, we observe that the lower bound, A^-, remains close to the sum of the area of the three ellipses. We can see that the gap increases first linearly, then exponentially

Table 5 Cutting ellipses with different eccentricity with ($\mathrm{EP}_{\mathrm{QP}}$). CPU time limit is 5 hours; GAMS 24.0.2; platform 1.

test case	ε	$\sum_i A_i$	**BARON** A^-	A^+	$10^3\Delta$	**LindoGlobal** A^-	A^+	$10^3\Delta$	**GloMIQO** A^-	A^+	$10^3\Delta$	$\frac{\sum_i A_i}{A^+}$
TE1.00	0.0000	15.61372	***	22.17171	0.01	***	22.17171	0.0001	***	22.17171	0.01	70.42181%
TE0.99	0.1411	15.45758	21.73027	21.84219	5.15	21.73049	21.84169	5.12	21.73027	21.84169	5.13	70.77099%
TE0.98	0.1990	15.30144	21.29350	21.50833	10.09	21.29371	21.50937	10.13	21.29350	21.50833	10.09	71.14193%
TE0.97	0.2431	15.14530	20.86115	21.17669	15.13	20.86136	21.17669	15.12	20.86115	21.17669	15.13	70.41597%
TE0.96	0.2800	14.98917	20.43324	20.85335	20.56	20.43345	20.84672	20.23	20.43324	20.84672	20.24	71.90182%
TE0.95	0.3122	14.83303	20.00977	20.51837	25.42	20.00997	20.51837	25.41	20.00977	20.51837	25.42	72.29146%
TE0.90	0.4359	14.05234	17.95890	18.89960	52.38	17.95908	18.90025	52.41	17.95890	18.89960	52.38	74.35258%
TE0.80	0.6000	12.49097	14.18974	16.09992	134.62	14.18989	16.09992	134.60	14.18975	16.09992	134.62	77.58405%
TE0.70	0.7141	10.92960	10.92949	13.79909	262.56	11.63246	13.79909	186.26	10.92949	13.79909	262.56	79.20522%
TE0.60	0.8000	9.36823	9.36814	11.65005	243.58	9.36823	11.65005	243.57	9.36814	11.65005	243.58	80.41365%
TE0.50	0.8660	7.80686	7.80678	9.74384	248.13	7.80686	9.74384	248.11	7.80678	9.74384	248.13	80.12098%
TE0.40	0.9165	6.24549	6.24542	7.91654	267.57	6.24549	7.91654	267.56	6.24542	7.91654	267.58	78.89166%
TE0.30	0.9539	4.68411	4.68407	6.16566	316.30	4.68412	6.16566	316.29	4.68407	6.16566	316.30	75.97094%
TE0.20	0.9798	3.12274	3.12271	4.15789	331.50	3.12274	4.15807	331.55	3.12271	4.15789	331.50	75.10396%
TE0.10	0.9950	1.56137	1.56136	2.08193	333.41	1.56137	2.08193	334.40	1.56136	2.08193	333.41	74.99628%

*** solved to proven global optimality (within 10^{-5})



(a) TE0.80: $A^+ = 16.09992$ (b) TE0.50: $A^+ = 9.74384$

(c) TE0.10: $A^+ = 2.08193$

Fig. 6 Feasible ellipse cuttings computed via ($\mathrm{EP}_{\mathrm{QP}}$) for three ellipses with different eccentricities.

and then remains approximately stable, as ρ decreases from 1.0 to 0.1; *cf.* Fig. 5. Three ellipse cuttings for $\rho = 0.80, \rho = 0.50$, and $\rho = 0.10$ are plotted in Fig. 6. Though we realize that the gaps are not closed (for $\rho < 1$), the ratio of the area of the rectangle to the areas of the ellipses appears to stay above 70% and peaks at approximately 80.5% for $\rho \approx 0.60$.

4.2.3 Identical Ellipses

Before we report on our numerical results using our mathematical programming formulations to cut sets of identical ellipses, we derive quasi-analytic cutting configurations which we will call symmetric for simplicity. This allows us to test our formulations, benchmark the global solvers and compare the solution to unsymmetrical cuttings.

For identical ellipses, we can place the ellipses symmetrically, enabling us to compute feasible cuttings as follows. For the case of three ellipses, we place them as shown in Fig. 7(a). The three ellipses are then centered and orientated at $(2,1;0°),(2,3;0°)$ and $(x^0_{31},2;90°)$ with nomenclature $(x^0_{i1},x^0_{i2};\theta_i)$. Let us refer to this configuration as $2-\theta_i$ which means: two ellipses with $\theta_i=0°$, and one ellipse with rotation angle $\theta_i=90°$. The results we derive provide upper bounds as long the width of the rectangle is $x^R_2\geq 4$. They are expected to be exact for $x^R_2=4$.

Ellipse 1 and 3 touch each other at point (x_s,y_s) which can be obtained by solving the following optimization problem:

$$\begin{aligned}
\max\ & x^0_{31} \\
\text{s.t.}\ & \frac{(x_s-2)^2}{4}+\frac{(y_s-1)^2}{1}=1 && (63)\\
& \frac{\left(x_s-x^0_{31}\right)^2}{1}+\frac{(y_s-2)^2}{4}=1 && (64)\\
& x^0_{31}>x_s \\
& 0.5\leq y_s\leq 1.5
\end{aligned}$$

Resolving (63) and (64) leads to equation

$$x_s=2\sqrt{2y_s-y_s^2}+2=x^0_{31}-\sqrt{1-\tfrac{1}{4}\left(y_s-2\right)^2}\quad ,$$

which allows us to write x^0_{31} as a function of y_s

$$x^0_{31}=x^0_{31}(y_s)=2\sqrt{2y_s-y_s^2}+2+\sqrt{1-\tfrac{1}{4}\left(y_s-2\right)^2}\quad .$$

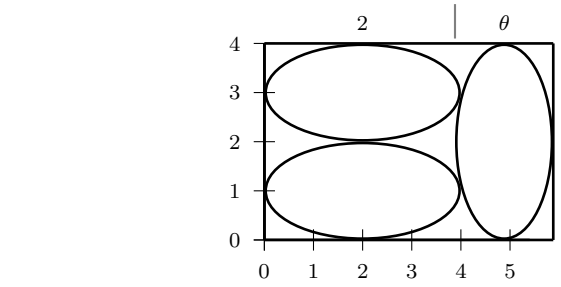
Its maximum is obtained by equating the first derivative to zero

$$\frac{2-2y_s}{\sqrt{2y_s-y_s^2}}+\frac{2-y_s}{4\sqrt{1-\frac{1}{4}\left(y_s-2\right)^2}}=0\quad , \qquad (65)$$

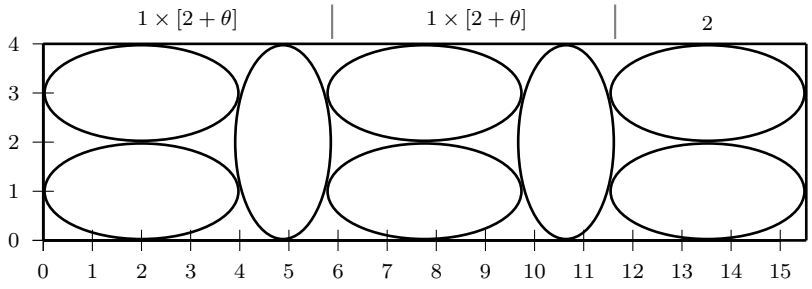
and yields the numerical values $(y_s,x^0_{31})\approx(1.121357,4.88355)$; (65) does not possess an analytical solution. For completeness, we also report $x_s\approx 3.98521$ and the area,

$$\tilde{a}_3:=4x^{\mathrm{R}}_{1,3}:=4\left(x^0_{31}+1\right)\approx 23.53419\quad .$$

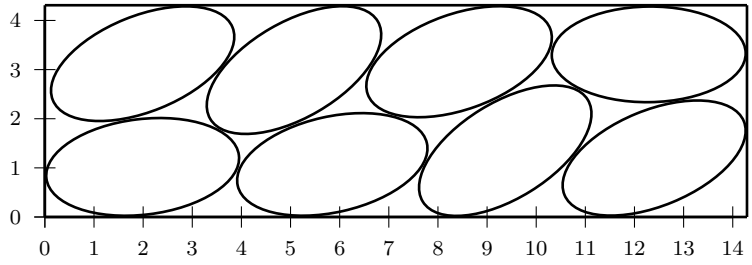
The values of x^0_{31} and $x^{\mathrm{R}}_{1,3}$ can also be used to derive good upper bounds, $\tilde{a}_n=4x^{\mathrm{R}}_{1,n}$, on the area of rectangles for n identical ellipses.



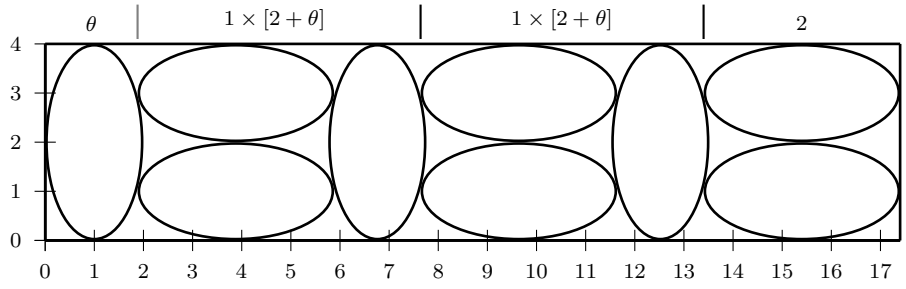
(a) "symmetric:" $A^+ = 23.53419$



(b) "symmetric:" $A^+ = 62.13676 = 15.53419 \cdot 4$



(c) "asymmetric:" $A^+ = 61.62058 \approx 14.28642 \cdot 4.31323$



(d) "symmetric:" $A^+ = 69.67096$

Fig. 7 Ellipse cuttings for identical ellipses.

Table 6 Cuttings obtained for "symmetric" placing of ellipses.

n	configuration C_n	recursion	$x^{\mathrm{R}}_{1,n}$	$\tilde{a}_n = 4x^{\mathrm{R}}_{1,n}$
3	$2+\theta = 1 \times [2+\theta]$		x_3+1	23.53419
4	$\theta + C_3 = \theta + 1 \times [2+\theta]$	$\theta + C_3$	$2(x_3 - 2 + 1)$	31.06838
5	$2+\theta+2 = 1 \times [2+\theta] + 2$	$C_3 + 2$	$2x_3$	39.06838
6	$\theta + 1 \times [2+\theta] + 2$	$\theta + C_5$	$x^{\mathrm{R}}_{1,4} - 1 + x_3$	46.60257
7	$\theta + 2 \times [2+\theta]$	$C_6 + \theta$	$2(x^{\mathrm{R}}_{1,4} - 1)$	54.13676
8	$2 \times [2+\theta] + 2$	$2 + C_6$	$2(x^{\mathrm{R}}_{1,5} - 2)$	62.13676
9	$\theta + 2 \times [2+\theta] + 2$	$C_7 + 2 = \theta + C_6$	$x^{\mathrm{R}}_{1,7} - 1 + x_3$	69.67096
10	$\theta + 3 \times [2+\theta]$	$C_9 + \theta$	$2(x^{\mathrm{R}}_{1,6} - 2)$	77.20515
11	$3 \times [2+\theta] + 2$	$2 + C_9$	$2(x^{\mathrm{R}}_{1,6} - 1)$	85.20515
12	$\theta + 3 \times [2+\theta] + 2$	$C_{10} + 2 = \theta + C_{11}$	$x^{\mathrm{R}}_{1,10} - 1 + x_3$	92.73934
13	$\theta + 4 \times [2+\theta]$	$C_{12} + \theta$	$2(x^{\mathrm{R}}_{1,7} - 1)$	100.27353
14	$4 \times [2+\theta] + 2$	$2 + C_{12}$	$2(x^{\mathrm{R}}_{1,8} - 2)$	108.27353
15	$\theta + 4 \times [2+\theta] + 2$	$\theta + C_{14}$	$x^{\mathrm{R}}_{1,13} - 1 + x_3$	115.80772

For $n \geq 7$, a general formula is

$$x^{\mathrm{R}}_{1,n} = \left(x^{\mathrm{R}}_{1,n-2} - 1\right) + x_3 \quad ,$$

with $x_3 \equiv x^0_{31} \approx 4.88355$ as derived for the three ellipse case. If the configuration is symmetric, we have

$$x^{\mathrm{R}}_{1,n} = 2 \begin{cases} x^{\mathrm{R}}_{1,1+\lfloor n/2 \rfloor} - 1, & \text{for } \; n = 2k + (k-1) = 3k - 1 \wedge k = 2m \\ x^{\mathrm{R}}_{1,1+\lfloor n/2 \rfloor} - 2, & \text{for } \; n = 2k + (k-1) = 3k - 1 \wedge k = 2m+1 \\ x^{\mathrm{R}}_{1,1+\lfloor n/2 \rfloor} - 2, & \text{for } \; n = 2k + (k+1) = 3k + 1 \wedge k = 2m+1 \\ x^{\mathrm{R}}_{1,1+\lfloor n/2 \rfloor} - 1, & \text{for } \; n = 2k + (k+1) = 3k + 1 \wedge k = 2m \end{cases}$$

We summarize the resulting symmetric cuttings in Table 6. Fig. 7(a), (b) and (d) illustrate the notation used in Table 6 for 3, 8 and 9 ellipses, respectively.

We start with a series of computations for identical ellipses using the monolith formulation ($\mathrm{EP}_{\mathrm{QP}}$), as summarized in Table 7. With only 20 minutes of CPU time, the lower bounds do not increases above the sum of the area of all ellipses. This picture does not change when allowing 45 minutes. However, the runs terminated after 45 minutes tend to reveal improved solutions compared to the shorter runs of 20 minutes. Note that the lower bounds obtained by LindoGlobal are slightly larger when compared to those obtained by the other two solvers.

As long as the upper bound on x^P_2 equals 4 (test cases TS02, TS03, TS04), $\tilde{a}_n$ yields the optimal area of the design rectangle. They are obtained by the three solvers (*cf.* Table 7). For $x^R_2 > 4$ (all other test cases), the solutions obtained are often slightly better than the analytic bounds $\tilde{a}$ because additional topological placements to the symmetric ones become possible. We observe this the first time for TC08. Fig. 7(b) and (c) contrast the "symmetrical" to the better "asymmetrical" placement.

Table 7 Identical ellipses with (EP_{QP}) (no MINLP extensions). For 16 and more identical ellipses, no feasible points were found by any of the three solvers used. GAMS 24.1; platform 2.

test case	$\tilde{a}_n$	**BARON** A^-	**BARON** A^+	**LindoGlobal** A^-	**LindoGlobal** A^+	**GloMIQO** A^-	**GloMIQO** A^+
CPU time limit: 20 minutes							
TS02	16.00000	***	16.00000	15.99262	16.00000	***	16.00000
TS03	23.53419	18.84937	23.53383	18.84956	23.53419	18.84937	23.53418
TS04	31.06838	25.13249	31.06838	25.13274	31.06838	25.13249	31.06838
TS05	39.06838	31.41561	44.27202	31.41593	39.03709	31.41561	39.03440
TS06	46.60257	37.69873	46.59140	37.69911	46.87235	37.69873	46.59298
TS07	54.13676	43.98186	54.13676	43.98230	63.32598	43.98186	54.13676
TS08	62.13676	50.26498	61.26671	50.26548	61.26671	50.26498	61.26671
TS09	69.67096	56.54810	69.58410	56.54867	69.58592	56.54810	69.58410
TS10	77.20515	62.83122	76.49471	62.83185	78.23977	62.83122	76.49471
TS11	85.20515	69.11435	85.73779	69.11504	–	69.11435	84.61446
TS12	92.73934	75.39747	91.67122	75.39822	94.22207	75.39747	91.67122
TS13	100.27353	81.68059	99.85158	81.68141	–	81.68059	99.85158
TS14	108.27353	87.96371	106.78443	87.96459	–	87.96371	110.6829
TS15	115.80773	94.24684	115.13250	94.24778	–	94.24684	–
CPU time limit: 45 minutes							
TS02	16.00000	***	16.00000	***	16.00000	***	16.00000
TS03	23.53419	18.84942	23.53383	18.84956	23.53351	18.84956	23.53418
TS04	31.06838	25.13249	31.06838	25.13274	31.06838	25.13274	31.06838
TS05	39.06838	31.41561	39.01646	31.41593	39.03709	31.41561	39.01646
TS06	46.60257	37.69873	46.59133	37.69911	46.87236	37.69873	46.59133
TS07	54.13676	43.98186	54.13676	43.98230	63.32598	43.98186	54.13676
TS08	62.13676	50.26498	61.26671	56.00000	68.00000	50.26498	61.26671
TS09	69.67096	56.54810	69.58409	56.54867	69.58410	56.54810	69.58410
TS10	77.20515	62.83122	76.49471	62.83185	76.49544	62.83122	76.49471
TS11	85.20515	69.11435	85.64764	69.11504	87.87180	69.11435	84.61819
TS12	92.73934	75.39747	91.67122	75.39822	94.22207	75.39747	91.67122
TS13	100.27353	81.68059	99.85158	81.68141	108.91049	81.68059	99.85158
TS14	108.27353	87.96371	106.78443	87.96459	–	87.96371	106.9077
TS15	115.80772	94.24684	115.13250	94.24778	–	94.24684	–

*** solved to proven global optimality (within 10^{-5})
– no feasible solution found

Computations exploiting the lexicographic ordering – the MINLP extension as discussed in Sect. 2.4 – are summarized in Table 8. For eight ellipses, the lower bounds are improved compared to the monolith formulation without the lexicographic ordering; *cf.* Table 7. Because the MINLP is more difficult to solve than the pure NLP formulation, we are not surprised to see the upper bound weaker. However, the advantage of the lexicographic ordering becomes obvious when we try to close the gap; at least for three ellipses.

On platform 3, GAMS 23.8, GloMIQO, and the lexicographic MINLP approach, the gap is closed after 128,329 seconds (35h 38m 49s) within a tolerance of $\Delta = 10^{-5}$. The global solution has objective function value $a = 23.53347$. The gap could neither be closed using GloMIQO embedded in GAMS 23.9 to GAMS 24.1, nor by the other two global solvers, with 48 hours of computational

Table 8 Identical ellipses with (EP_{QP}) exploiting lexicographic ordering. CPU time limit is 45 minutes; GAMS 24.1; platform 2.

test case	$\tilde{a}$	BARON		LindoGlobal		GloMIQO	
		A^-	A^+	A^-	A^+	A^-	A^+
TS02	16.00000	***	16.00000	***	16.00000	***	16.00000
TS03	23.53419	18.84937	23.53383	18.84992	23.53305	18.84937	23.53418
TS04	31.06838	25.13249	31.06838	25.13274	31.06838	25.13249	31.06838
TS05	39.06838	31.41561	44.27205	31.41593	39.03709	31.41561	39.01646
TS06	46.60257	37.69873	–	37.69911	46.87236	37.69873	47.26726
TS07	54.13676	43.98186	–	43.98230	63.32598	43.98186	64.31777
TS08	62.13676	50.26498	–	50.26548	68.00000	56.12506	67.47646
TS09	69.67096	56.54810	–	56.54867	–	56.54810	78.41787
TS10	77.20515	62.83122	–	62.83185	–	62.83122	84.86555

*** solved to proven global optimality (within 10^{-5})
– no feasible solution found

time. BARON computed a cutting with area 23.53383; *cf.* Table 7. It turns out that this is smaller than the lower bound obtained by GloMIQO. The explanation lies in the different feasibility tolerances of the two global solvers.

4.3 Polylithic

For the larger test instances (TC20, TC30, TC50 and TC100), none of the global or local NLP solvers available in GAMS can compute a feasible point. Therefore, we analyze these cases using our polylithic approaches.

We choose value 10 for parameter n_1 for both heuristics, because we can obtain reasonable solutions within approximately 5 to 15 minutes. The second parameter is n_2, the number of ellipses placed during the sequential phase. We experienced with n_2=1 and n_2=2. While, from the underlying mathematical idea, we expect that the solutions are better for n_2=2, the computing effort is higher. This in turn, can lead – and sometimes does – to worse solutions if the time limit is reached before the gap is closed. Thus, considering this trade-off, we recommend and prefer $n_2 = 1$.

We benchmark the polylithic approach by comparing it to the best results obtained by the monolith one (Table 3) for TC11 and TC14. For TC11, the monolith yields $A^- = 47.31238$ and $A^+ = 64.59177$ (LindoGlobal) whereas H1 provides $A^- = 37.11006$ and $A^+ = 57.24034$ (BARON) and H2 provides $A^+ = 57.73518$ (BARON) and for TC14, the monolith yields $A^- = 20.61650$ and $A^+ = 29.65886$ (GloMIQO) whereas H1 provides $A^- = 13.69538$ and $A^+ = 24.67185$ (BARON) and H2 provides $A^+ = 24.84634$ (GloMIQO). Thus, for TC11 and TC14 both heuristics find better cuttings than the monolith formulation.

We reach the limits of H1 when we approach 30 and more ellipses (the solvers cannot compute a feasible cutting for the resulting (EP) in step H1_1.2.4). The front heuristics, H2, works fine for all cases TC11, TC14, TC20, TC30,

Table 9 Heuristic methods H1 ($n_1 = 10$) and H2 ($n_1 = 10$, $n_3 = 5$) for cutting ellipses; $|S| = 8$. CPU time limit is 5 hours (for steps H1_1.1, H1_1.2.4, H1_1.2.5, H2_1.2 and H2_1.2.4); GAMS 24.0.2; platform 1.

test case	BARON A^-	seq.	BARON A^+	seq.	LindoGlobal A^-	seq.	LindoGlobal A^+	seq.	GloMIQO A^-	seq.	GloMIQO A^+	seq.
H1: $n_2 = 1$												
TC11	37.11006	2	57.24034	3	37.11006	2	58.34738	5	37.11006	2	58.30057	5
TC14	13.69538	2	24.67185	1	13.69538	3	24.86237	4	13.69538	2	25.12648	2
TC20	24.92068	2	69.17487	3	mtol		mtol		24.92068	4	67.95753	2
H1: $n_2 = 2$												
TC11	37.11006	2	57.24034	3	37.11006	2	58.30057	5	37.11006	2	58.34738	5
TC14	11.04466	2	24.97945	6	13.69538	2	25.93243	1	11.04466	2	25.12648	5
TC20	24.92068	2	69.51981	4	mtol		mtol		19.86272	4	69.82890	7
H2: $n_2 = 1$												
TC11	n/a		57.73518	7	n/a		58.34738	5	n/a		58.30057	5
TC14	n/a		25.67342	1	n/a		25.72716	4	n/a		24.84634	8
TC20	n/a		68.47550	4	n/a		68.23159	2	n/a		67.83459$^\natural$	1
TC30	n/a		103.45212	2	n/a		113.05175	6	n/a		109.43025$^\natural$	1
TC50	n/a		167.10549	4	n/a		176.89313	7	n/a		174.43153$^\natural$	1
TC100	n/a		326.64228	1	n/a		–		n/a		331.77321$^\natural$	1
H2: $n_2 = 2$												
TC11	n/a		57.73518	7	n/a		58.34738	5	n/a		58.30057	5
TC14	n/a		25.72871	2	n/a		26.75323	2	n/a		25.73702	5
TC20	n/a		71.14951	7	n/a		–		n/a		78.31584$^\natural$	1
TC30	n/a		104.31177	7	n/a		–		n/a		105.57857$^\natural$	1
TC50	n/a		167.71486	8	n/a		–		n/a		166.91505$^\natural$	1
TC100	n/a		325.23287	3	n/a		–		n/a		322.64663$^\natural$	1

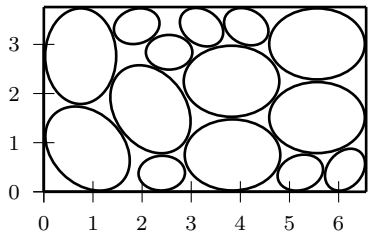
–	no feasible solution found in any of the sequences
$\natural$	CPU time limit is 1 hour; GAMS 24.0; platform 2
n/a	H2 does not provide a lower bound on the area of the design rectangle
mtol	model too large for the available licence

TC50 and TC100. For both H1 and H2, results with $n_2 = 2$ are superior to $n_2 = 1$ but are somewhat more computational expensive. Computations for $n_2 = 3$ are even more challenging and do not provide a clear direction.

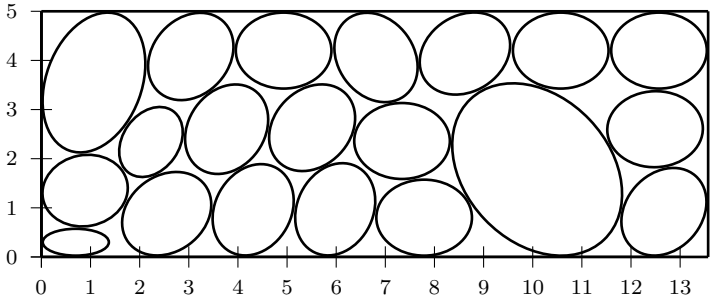
Fig. 8(a) shows a feasible cutting computed by BARON with H1 ($n_2 = 1$) and Fig. 8(b) the cutting computed by GloMIQO with H2 ($n_2 = 1$).

5 Conclusions

We developed non-convex (MI)NLP models describing the problem of cutting ellipses from a design rectangular plate. Small problem instances can be solved with the current state-of-the art global solvers available in GAMS. The more the ellipses deviate from circles, the more difficult it is to close the gap. As it is expected from the NP-hard nature of the ellipse cutting problem, global solvers reach their limitations fast and it becomes a very challenging task for the solvers just to compute a feasible point. For these cases, we have developed two polylithic methods, generating good ellipse cuttings.



(a) TC14: $A^+ = 24.67185$



(b) TC20: $A^+ = 67.83459$

Fig. 8 Feasible cuttings computed by the heuristics.

The developed (MI)NLP formulations lean themselves naturally to higher dimensional extensions. The 3-D case is of particular practical interest. Who was never curious on how to pack smarties, a German chocolate sweet, optimally?

Acknowledgements Thanks are directed to Prof. Dr. Siegfried Jetzke (Ostfalia Hochschule, Salzgitter, Germany) for his interest in this work, comments on the manuscript and discussion about the usefulness of ellipses and ellipsoids in real world problems.

A Notation

We start with the notation used in the derivation of the model; they are not used in the mathematical programming formulations directly.

$\mathsf{A}_{\theta i}$	positive definite matrix defining ellipses; entries are A_{11}, A_{12}, A_{21}, and A_{22}
$\mathbf{c}$	objective function coefficient vector of auxiliary problems; $\mathbf{c}^\top = (1,0)$ or $\mathbf{c}^\top = (0,1)$
d_i	distance of ellipse i to line $\mathbf{G}(t)$
D_i	diagonal matrix for ellipse i with eigenvalues of $\mathsf{A}_{\theta i}$ in the diagonal
δ_{ij}^{ab}	maximum vertical extension of ellipse i to the hyperplane in the new coordinate system
δ_{ij}^{be}	maximum vertical extension of ellipse j to the hyperplane in the new coordinate system

$\mathsf{E}_{\theta-\omega,i}$	positive definite matrix defining rotated ellipses; $\mathsf{E}_{\theta-\omega,i} := \mathsf{A}_{\theta-\omega,i}$; entries are E_{11}, E_{12}, E_{21}, and E_{22}
$\mathbf{G}(t)$	separating line, *i.e.*, hyperplane
$\mathcal{L}(\mathbf{x}, \bar{\lambda})$	Lagrangian function
λ_{id}	eigenvalue of matrix $\mathsf{A}_{\theta i}$; $\lambda_{i1} = a_i^{-2}$ and $\lambda_{i2} = b_i^{-2}$
$\bar{\lambda}$	Lagrangian multiplier associated with ellipse equation
φ_i	rotation angle to generate ellipse i in new coordinate system
$\mathsf{R}_{\theta i}$	rotation matrix for ellipse i at angle θ_i
$\mathbf{v}_i(\varphi_i)$	equation for ellipse i in new coordinate system
v_{id}^0	center coordinate (in dimension d) of ellipse i in the new coordinate system
v_{i2}^-	minimal extension (in dimension $d = 2$) of ellipse i in the new coordinate system
x_{id}^-	minimum extension of ellipse i in dimension d
x_{id}^+	maximum extension of ellipse i in dimension d

The notation used in the mathematical programming models and heuristics is summarized in the following sections.

A.1 Indices and Sets

$d \in \{1, 2\}$	index for the dimension; $d = 1$ represents the length and $d = 2$ the width
$i \in \mathcal{I} := \{1, \ldots, n\}$	objects (ellipses or circles) to be cut
$(c_x, c_y) \in \mathcal{I}^{\text{ce}}$	small rectangles, cells, dividing the design rectangle
$(i, j) \in \mathcal{I}^{\text{co}}$	pairs of congruent ellipses; we assume $i < j$

A.2 Data

a_i	semi-major axis of ellipse i; $a_i \geq b_i$
$\tilde{a}_n$	area of the design rectangle for the "symmetric" cutting for n identical ellipses
A_i	area of ellipse i; $A_i = \pi a_i b_i$
A^-, A^+	lower and upper bound on the area, a, of the design rectangle obtained during the computation
$A^{\text{ci},-}$	minimal area of the design rectangle to host the inner circles associated with the ellipses. $A^{\text{ci},-}$ provides a lower bound on the associated ellipse cutting problem
$A^{\text{ci},+}$	area of the design rectangle to host the outer circles associated with the ellipses. $A^{\text{ci},-}$ provides an upper bound on the associated ellipse cutting problem
b_i	semi-minor axis of ellipse i; $a_i \geq b_i$
$C^-_{c_x c_y d}$, $C^+_{c_x c_y d}$	lower and upper boundary coordinate of cell (c_x, c_y)
$\overline{D}_{ij}$	bound on the distance variables d_{ij}^{ab} and d_{ij}^{be}
Δ	relative gap
ε	eccentricity measuring the deviation of an ellipses from a circle
n_1, n_2, n_3	parameters in the heuristics: number of ellipses selected
R_i	radius of circle i to be cut

ρ	factor (parameter) for semi-minor axis for test instances TEρ
S_{id1}, S_{id2}	auxiliary data derived from semi-major and semi-minor axis, as defined in (36)
S_d^-, S_d^+	minimum (lower bound) and maximum size (upper bound) of the extension of the design rectangles in dimension d
$x_{1,n}^{\mathrm{R}}$	length of the design rectangle for the "symmetric" cutting of n identical ellipses
X^-	smallest center coordinate among the n_3 ellipses in coordinate direction $d = 1$ – only used for heuristic H2

A.3 Decision Variables

a	(continuous) area of the design rectangle; a^* defines (globally) optimal area
d_{ij}^{ab}	(continuous) distance of the center of ellipse i to the separating line between the ellipses i and j; ellipse i is above the separating line
d_{ij}^{be}	(continuous) distance of the center of ellipse j to the separating line between the ellipses i and j; ellipse j is below the separating line
$\delta_{ic_xc_y}$	(binary) assign ellipse i to cell (c_x, c_y)
g_{ijd}^0	(continuous) footpoint coordinate (in dimension d) of separating line between ellipses i and j; we place the footpoint inside the design rectangle; index ij is dropped in Sect. 2.1.4
g_{ijd}	(continuous) slope (in dimension d) of the separating line between ellipses i and j; index ij is dropped in Sect. 2.1.4
λ_{ij}	(continuous) linear combination variable for hyperplane and ellipses i and j
ω_{ij}	(continuous) inclination angle of the separating line between the ellipses i and j; $\omega_{ij} \in [0, 2\pi]$; index ij is dropped in Sect. 2.1.4
p_{ij}^{ab}	(continuous) auxiliary variable modeling $\cos(\theta_i - \omega_{ij})$; $p_{ij}^{\mathrm{ab}} \in [-1, 1]$
p_{ij}^{be}	(continuous) auxiliary variable modeling $\cos(\theta_j - \omega_{ij})$; $p_{ij}^{\mathrm{be}} \in [-1, 1]$
θ_i	(continuous) orientation angle of ellipse i; $\theta_i \in [0, 2\pi]$
v_i	(continuous) auxiliary variable representing trigonometric term $\cos\theta_i$; $v_i \in [-1, 1]$
w_i	(continuous) auxiliary variable representing trigonometric term $\sin\theta_i$; $w_i \in [0, 1]$
x_d^{R}	(continuous) extension of the design rectangle in dimension d
x_{id}^0	(continuous) coordinates of the center vector of object i to be cut
z	(continuous) waste of the design rectangle; $z = a - \sum_{i \in \mathcal{I}} A_i$

References

1. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global Optimization of Mixed Integer Nonlinear Problems. AIChE Journal **46**, 1769–1797 (2000)
2. Adjiman, C.S., Androulakis, I.P., Maranas, C.D., Floudas, C.A.: A Global Optimization Method aBB for Process Design. Computers & Chemical Engineering Suppl. **20**, S419–424 (1996)
3. Androulakis, I.P., Maranas, C.D., Floudas, C.A.: aBB: A Global Optimization Method for General Constrained Nonconvex Problems. Journal of Global Optimization **7**, 337–363 (1995)
4. Dyckhoff, H.: A Typology of Cutting and Packing Problems. European Journal of Operational Research **44**, 145–159 (1990)

5. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. Journal of Global Optimization **6**, 109–133 (1995)
6. Floudas, C.A.: (2012). Private communication
7. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J.: Global Optimization in the 21st Century: Advances and Challenges for Problems with Nonlinear Dynamics. Computers and Chemical Engineering **29**, 1185–1202 (2005)
8. Gensane, T., Honvault, P.: Optimal Packings of Two Ellipses in a Square. Tech. rep., L.M.P.A. (2012)
9. Hettich, R., Kortanek, K.O.: Semi-infinite Programming. SIAM Review **35**, 380–429 (1993)
10. Kallrath, J.: Combined Strategic Design *and* Operative Planning in the Process Industry. Computers and Chemical Engineering **33**, 1983–1993 (2009)
11. Kallrath, J.: Cutting Circles and Polygons from Area-Minimizing Rectangles. Journal of Global Optimization **43**, 299–328 (2009)
12. Kallrath, J.: Polylithic Modeling and Solution Approaches Using Algebraic Modeling Systems. Optimization Letters **5**, 453–466 (2011). 10.1007/s11590-011-0320-4
13. Liberti, L.: Reformulation and Convex Relaxation Techniques for Global Optimization. Ph.D. Thesis, Imperial College London, London, UK (2004)
14. Liberti, L., Pantelides, C.: An Exact Reformulation Algorithm for Large Nonconvex NLPs involving Bilinear Terms. Journal of Global Optimization **36**, 161–189 (2006)
15. Lindo Systems: Lindo API: User's Manual. Lindo Systems, Inc., Chicago (2004)
16. Lopez, M., Still, G.: Semi-infinite Programming. European Journal of Operational Research **180**, 491–518 (2007)
17. Maranas, C.D., Floudas, C.A.: Finding All Solutions of Nonlinearly Constrained Systems of Equations. Journal of Global Optimization **7**, 143–182 (1995)
18. Markót, M.C.: Interval Methods for Verifying Structural Optimality of Circle Packing Configurations in the Unit Square. J. Computational and Applied Mathematics **199**, 353–357 (2007)
19. Markót, M.C., Csendes, T.: A New Verified Optimization Technique for the "Packing Circles in a Unit Square" Problems. SIAM J. Optimization **16**, 193–219 (2005)
20. Miller, P.: Globally Optimal Packing of Nonconvex Two-Dimensional Shapes by Approximation with Ellipses. Senior Thesis, Princeton University, Princeton, NJ (2012)
21. Misener, R., Floudas, C.: GloMIQO: Global Mixed-integer Quadratic Optimizer. Journal of Global Optimization pp. 1–48 (2012). 10.1007/s10898-012-9874-7
22. Rebennack, S., Kallrath, J.: Continuous Piecewise Linear δ-Approximations for MINLP Problems. I. Minimal Breakpoint Systems for Univariate Functions. Tech. rep., Colorado School of Mines Division of Economcis and Business Working Paper Series 2012-12 (2012)
23. Rebennack, S., Kallrath, J.: Continuous Piecewise Linear δ-Approximations for MINLP Problems. II. Bivariate and Multivariate Functions. Tech. rep., Colorado School of Mines Division of Economcis and Business Working Paper Series 2012-13 (2012)
24. Rebennack, S., Kallrath, J., Pardalos, P.M.: Column Enumeration based Decomposition Techniques for a Class of Non-convex MINLP Problems. Journal of Global Optimization **43**, 277–297 (2009)
25. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. Nonconvex Optimization And Its Applications Series. Kluwer Academic Publishers, Dordrecht, The Netherlands (2002)