General Ellipse Packings in an Optimized Circle Using Embedded Lagrange Multipliers

Frank J. Kampas [a], János D. Pintér [b, c], Ignacio Castillo [d,*]

[a] Physicist at Large Consulting LLC, Ambler, PA, USA
[b] Pintér Consulting Services Inc. [c] Sobey School of Business, Saint Mary's University, Halifax, NS, Canada
[d] Lazaridis School of Business and Economics, Wilfrid Laurier University, Waterloo, ON, Canada
[*] Corresponding author. E-mail address: icastillo@wlu.ca

Abstract

The general ellipse packing problem is to find a non-overlapping arrangement of n ellipses with (in principle) arbitrary size and orientation parameters inside a given type of container set. Here we consider the general ellipse packing problem with respect to an optimized circle container with minimal radius. Following the review of selected topical literature, we introduce a new model formulation approach based on using embedded Lagrange multipliers. This optimization model is implemented using the computing system *Mathematica*: we present illustrative numerical results using the LGO global-local optimization software package linked to *Mathematica*. Our study demonstrates the applicability of the embedded Lagrange multipliers based modeling approach combined with global optimization tools to solve challenging ellipse packing problems.

1 Introduction and Review of Related Work

1.1 Circle Packings

In a general setting, a circle packing is an optimized non-overlapping arrangement of n arbitrary size circles inside a container (such as a circle, square or a general rectangle). The quality of the packing is typically measured by the size (area) of the container. The circle packing problem – in particular the case of identical circles – has received considerable attention as reflected by the literature. Due to the special (inherently symmetric) structure of this problem-type, studies dealing with identical circle packings often aim to prove the optimality of the configurations found, either theoretically or with the help of rigorous computational approaches: consult e.g. Szabó *et al.* (2001, 2005, 2007), Markót (2005) with numerous related further references therein.

The arbitrary sized circle packing problem is a significant generalization of the uniform sized case, since now each packed circle can have a different (in principle, arbitrary) radius. Generally speaking, provably optimal configurations can be found only to very small model instances ($n \leq 4$). Therefore studies dealing with general circle packings typically introduce and apply efficient generic or tailored global scope solution strategies, but without the proven optimality of the results obtained: cf. e.g. Riskin *et al.* (2003), Castillo and Sim (2004), Pintér and Kampas (2005, 2006a, b), Kampas and Pintér (2006), Addis *et al.* (2008), Castillo *et al.* (2008), Grosso *et al.* (2010).

Without going into further details related to circle packings, we refer to Castillo *et al.* (2008) and to Hifi and M'Hallah (2009) for reviews of both uniform and arbitrary sized circle packing problems and applications. Let us also remark that more general – and often very challenging – packing problem-types with a range of important real-world applications are discussed in the edited volume (Fasano and Pintér, 2015).

1.2 Ellipse Packings

The ellipse packing problem has received relatively little attention in the literature so far. Finding high quality (globally optimized) ellipse packings is a difficult computational problem, especially when dealing with packing ellipses of arbitrary size and orientation. The key challenge is the modeling and enforcement of the no-overlap constraints. Let us note here that measuring the overlap between two ellipses depends also on the orientation of the ellipses, in addition to the location of their centers.

Next, we briefly review some of the related literature. Even if not all works cited here are aimed at handling the exact same problem-type addressed by our present study, these works illustrate the significant difficulty of similar packings.

First we mention an exact result that deals with the densest packing of just two non-overlapping congruent ellipses in a square. In this case, for all real numbers r in $[0,1]$, Gensane and Honvault (2014) analytically describe the densest packing of two ellipses with aspect ratio r.

Birgin *et al.* (2013) study the problem of packing sets of identical circles within an ellipse. The basic challenge here is the closed formula based calculation to compute the distance of an arbitrary point to the boundary of the containing ellipse. The authors note that – even when considering only identical size circles – the resulting models are challenging nonlinear programming problems. In order to seek for globally optimized solutions, the authors propose stochastic multi-start and lattice-based search strategies.

Litvinchev *et al.* (2015) find optimized packings of "circular-like" objects – including circles, ellipses, rhombuses, and octagons – in a rectangular container. The authors propose a linear 0-1 model formulation based on a grid that approximates the container, and then consider the nodes of the grid as potential positions for assigning centers of the objects. The resulting binary linear programming problem is solved using the commercial software package CPLEX. Numerical results related to packing circles, ellipses, rhombuses, and octagons are presented. Let us point out that, given the grid approximation of the container, this approach can only handle the packing of uniform sized and orthogonally oriented ellipses inside a container: this is clearly a limitation in the context of our present study.

Galiev and Lisafina (2013) study the problem of packing uniform sized and orthogonally oriented ellipses inside a rectangular container. Similarly to Litvinchev et al. (2015), linear 0-1 model formulations are proposed using a grid that approximates the container. Two special cases regarding the orientation of the ellipses are considered: i) the major

axes of all the ellipses are parallel to the x or y axis, and ii) the major axes of some of the ellipses are parallel to the x axis and others to the y axis. A heuristic algorithm based on the linear model formulations is proposed and numerical results are presented.

Kallrath and Rebennack (2014) address the problem of packing ellipses of arbitrary size and orientation into an optimized rectangle (of minimal area). The packing model formulation is introduced as a cutting problem. The key idea is to use separating lines to ensure that the ellipses do not overlap with each other. For problem-instances with $n \leq 14$ ellipses, the authors present feasible solutions that are globally optimal subject to the finite arithmetic precision of the global solvers at hand. However – according to these authors – for $n > 14$ ellipses none of the local or global nonlinear optimization solvers available in conjunction with the GAMS modeling environment could compute a feasible solution. Therefore they propose heuristic approaches, in which the ellipses are added sequentially to an optimized rectangular container: this approach allows computing visibly high-quality solutions for up to 100 ellipses.

Uhler and Wright (2013) study the problem of packing arbitrary sized ellipsoids into an ellipsoidal container so as to minimize a measure of overlap between ellipsoids. A model formulation and two local scope solution approaches are discussed: one approach for the general case, and a simpler approach for the special case in which all ellipsoids are in fact spheres. The authors describe and illustrate their computational experience using chromosome organization in the human cell nucleus as the motivating application.

Based also on the illustrative references cited, we argue that ellipse packings have a number of practical applications, with a view also towards the future use of such models. Here we study the non-overlapping packing of ellipses with arbitrary size and orientation parameters inside a circular container: our objective is to minimize the radius of the container circle.

Packing ellipses into a circle requires i) the determination of the maximal distance from the center of the circular container to each ellipse (boundary), and ii) the finding of the minimal distance between all pairs of the ellipses. The first requirement is necessary to determine the radius of the circumscribing circle (which is then to be minimized). The second requirement is necessary to prevent the ellipses from overlapping. Explicit analytical formulas for these quantities – if they exist – would be complex. Therefore the approach taken here involves determining those quantities by embedding optimization calculations (using Lagrange multipliers) into the overall optimization strategy. In this Lagrangian setting, the optimization strategy to find the radius of the circumscribing circle and to prevent ellipse overlap proceeds simultaneously towards meeting both requirements. This approach allows us to solve the minimization problem numerically with a single call to a suitable global optimization procedure. While – analogously to the significantly easier case of general circle packings – we cannot guarantee the theoretical (provable) optimality of the ellipse configurations found, our work leads to visibly good quality ellipse packings. Packing three- or higher-dimensional ellipsoids can be seen as an immediate extension of the two-dimensional problem statement considered here.

Other related model-types and solution strategies will be discussed in our forthcoming studies.

2 Model Formulation

As stated above, the objective of the general ellipse packing problem studied here is to minimize the radius of the circumscribing circle. The inputs to the optimization problem are the semi-major and semi-minor axes of the ellipses to be packed. The primary decision variables are the radius of the circumscribing circle, and the centre position and orientation of each of the packed ellipses. Secondary (induced) variables are the positions of the points on the ellipses most distant from the center of the circumscribing circle, and the positions of the points on one of each pair of ellipses which minimizes the value of the equation describing the other ellipse. Other secondary variables are the embedded Lagrange multipliers used to determine those points. Note that all secondary variables are implicitly determined by the primary decision variables.

The constraints fall into two groups. The first constraint group uses the secondary variables to represent the constraints that keep the ellipses inside the circumscribing circle and prevent them from overlapping. The second constraint group represents the equations generated by the embedded Lagrange multiplier conditions. In our global optimization strategy, the calculations for finding the radius of the circumscribing circle and for preventing ellipse overlaps proceed simultaneously with the minimization of the radius of the circumscribing circle, rather than being performed to completion at each step towards the minimization of the radius.

Next, we present our formal model. Equation $eleq(a, b, xc, yc, \theta)(x, y)$ describes an ellipse with semi-major and semi-minor axes a and b, centered at $\{xc, yc\}$, and rotated counterclockwise by angle θ. More specifically, $eleq(a, b, xc, yc, \theta)(x, y)$ is negative for all points (x, y) inside the ellipse, zero for all points on the ellipse boundary, and positive for all points outside the ellipse. Note that a and b are given input parameters, while (xc, yc) and θ are primary decision variables for each ellipse i: the latter will be denoted by (xc_i, yc_i) and θ_i for $i = 1, \dots, n$.

Equation $eleq(a, b, xc, yc, \theta)(x, y)$ can be obtained by transforming the equation of a circle with radius 1, centered at (0,0), as follows.

$$\begin{aligned} eleq(a, b, xc, yc, \theta)(x, y) \\ = \left(\frac{\cos(\theta)\ (x - xc)}{a} + \frac{\sin(\theta)\ (y - yc)}{a}\right)^2 \\ + \left(\frac{\cos(\theta)\ (y - yc)}{b} - \frac{\sin(\theta)\ (x - xc)}{b}\right)^2 - 1 \end{aligned} \tag{1}$$

Note that in (1) the coordinate system is rotated by an angle of $-\theta$, which is equivalent to rotating the ellipse by an angle θ around its centre.

By assumption, the circumscribing circle is centered at the origin, so its radius must be at least the maximum value of $\sqrt{x^2+y^2}$ that can be obtained for all points (x,y) of the packed ellipses. The point on an ellipse with the maximum value of x^2+y^2 is clearly the same as the point which maximizes $\sqrt{x^2+y^2}$, and it can be determined using the Lagrange multiplier method by differentiating $x^2+y^2=\lambda\cdot el(x,y)$ with respect to x, y, and λ, where $el(x,y)$ represents the ellipse $eleq(a,b,xc,yc,\theta)(x,y)$. Applying this method, we obtain the equations

$$\begin{Bmatrix} 2x=\lambda\cdot el^{(1,0)}(x,y) \\ 2y=\lambda\cdot el^{(0,1)}(x,y) \\ el(x,y)=0 \end{Bmatrix}, \tag{2}$$

where $el^{(1,0)}(x,y)$ is the derivative of $el(x,y)$ with respect to x and $el^{(0,1)}(x,y)$ is the derivative of $el(x,y)$ with respect to y.

The next equation follows simply from the requirement that the point sought lies on the ellipse boundary. Note that λ can be eliminated from the first two equations: hence, we obtain

$$y\cdot el^{(1,0)}(x,y)=x\cdot el^{(0,1)}(x,y). \tag{3}$$

Since the slope of the ellipse boundary at point (x,y) is given by

$$\frac{dy}{dx}=\frac{\partial el(x,y)/\partial x}{\partial el(x,y)/\partial y}=\frac{el^{(1,0)}(x,y)}{el^{(0,1)}(x,y)}=\frac{x}{y} \tag{4}$$

from equation (3), the slope of the line from $(0,0)$ to (x,y), which is y/x , is the inverse of the slope of the ellipse at the point most distant from the origin. In other words, the line from the origin to the point on the ellipse most distant from the origin is perpendicular to the ellipse boundary at that point, as one might expect.

It is useful to setup equations for the derivatives of the ellipse equation with respect to x and y. These derivatives are:

$$\begin{aligned} &eleqdx(a,b,xc,yc,\theta)(x,y) \\ &\quad=\frac{2}{a^2b^2}(b^2(x-xc)\cos(\theta)^2 \\ &\quad-(a^2-b^2)(y-yc)\cos(\theta)\sin(\theta) \\ &\quad+a^2(x-xc)\sin(\theta)^2); \end{aligned} \tag{5}$$

$$\begin{aligned} &eleqdy(a,b,xc,yc,\theta)(x,y)=\frac{2}{a^2b^2}(a^2(y-yc)\cos(\theta)^2- \\ &(a^2-b^2)(x-xc)\cos(\theta)\sin(\theta)+b^2(y-yc)\sin(\theta)^2). \end{aligned} \tag{6}$$

The equations shown below are used to find the points that are closest and most distant from the origin. To obtain the most distant point, λ must be positive, since increasing the size of the ellipse increases the value of the maximum distance only, assuming that the center of the circle $(0,0)$ does not lie inside the ellipse. If this is the case, then other ellipses outside that particular ellipse will determine the radius of the container circle.

$$\begin{Bmatrix} 2x = \lambda \cdot eleqdx(a,b,xc,yc,\theta)(x,y) \\ 2y = \lambda \cdot eleqdy(a,b,xc,yc,\theta)(x,y) \\ eleq(a,b,xc,yc,\theta)(x,y) = 0. \end{Bmatrix} \tag{7}$$

For example, consider the ellipse defined by $eleq(1.25,0.75,1,2,\pi/3)(x,y)$. The point on it that is most distant from the origin can be found solving equations (7): the numerical solution is $(x,y) = (1.608,3.092)$, with corresponding distance value 3.485.

In the optimization strategy, the requirement on the (positive) sign of λ will be enforced by setting search bounds, rather than specifying a constraint. Moreover, the value of the maximum distance from the origin is obtained by evaluating $\sqrt{x^2+y^2}$ at the solution of equation (7). Note that equation (7) may fail, if the ellipse contains the origin. To handle this potential issue, constraints are added to the optimization strategy in order to keep the maximum distance point further from the origin than the smaller of the semi-major or semi-minor axis of the ellipse in question. For illustration, a packed ellipse, a possible circumscribing circle, and the point of their intersection are shown in Figure 1.

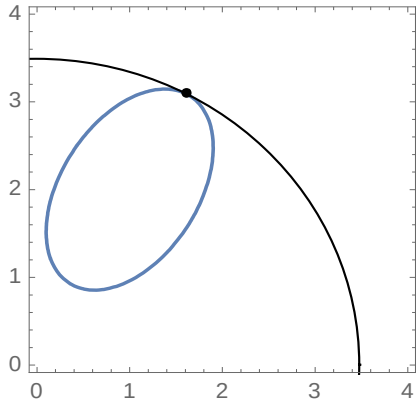

Figure 1. A packed ellipse, the circumscribing circle, and their intersection point

Proceeding now to prevent ellipse overlaps, all pairs of packed ellipses are prevented from overlapping by requiring that the minimum value of the ellipse equation for the first ellipse (say ellipse i), for any point on the second ellipse (say ellipse j), is greater than a judiciously set (sufficiently small) $\varepsilon \geq 0$. This requirement will also be accomplished using the embedded Lagrange multiplier method.

$$\begin{Bmatrix} el_i^{(1,0)}(x,y) = \lambda \cdot el_j^{(1,0)}(x,y) \\ el_i^{(0,1)}(x,y) = \lambda \cdot el_j^{(0,1)}(x,y) \\ el_j(x,y) = 0. \end{Bmatrix} \tag{8}$$

The last equation type is the requirement that the point lies on ellipse j. Eliminating λ from the first two equations, we obtain

$$el_i^{(0,1)}(x,y) \cdot el_j^{(1,0)}(x,y) = el_j^{(0,1)}(x,y) \cdot el_i^{(1,0)}(x,y). \tag{9}$$

Note that, as expected, the slope of the expanded ellipse i equals the slope of ellipse j at the point on ellipse j that minimizes or maximizes the value of the function describing ellipse i.

The equations shown below determine the point on ellipse j that maximizes or minimizes the value of the function describing ellipse i. In the case considered here, λ must be negative to obtain the minimum. As indicated before, in the optimization strategy the requirement on the sign of λ will be enforced by setting its search bounds rather than specifying an additional constraint.

$$\begin{Bmatrix} eleqdx(a_i,b_i,xc_i,yc_i,\theta_i)(x,y) = \lambda \cdot eleqdx(a_j,b_j,xc_j,yc_j,\theta_j)(x,y) \\ eleqdy(a_i,b_i,xc_i,yc_i,\theta_i)(x,y) = \lambda \cdot eleqdy(a_j,b_j,xc_j,yc_j,\theta_j)(x,y) \\ eleq(a_j,b_j,xc_j,yc_j,\theta_j)(x,y) = 0. \end{Bmatrix} \tag{10}$$

For example, consider the ellipses defined by $eleq_i(1.25,0.75,1,2,\pi/3)(x,y)$ and $eleq_j(1.5,0.83,-0.5,1,\pi/4)(x,y)$. The overlap (value of $eleq_i$) between the ellipses is -0.886 with $(x,y) = (0.701,1.777)$: this can be found by solving equation (10). Figure 2 shows this overlapping configuration.

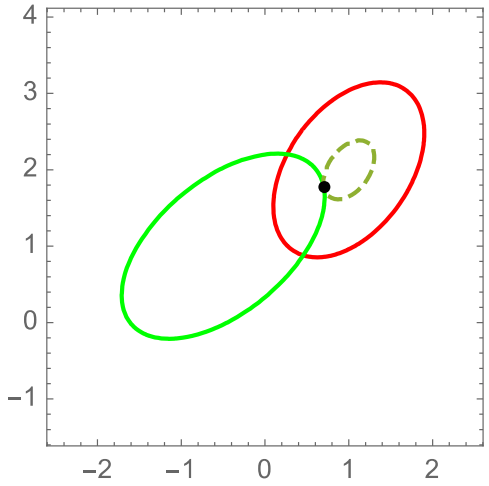

Figure 2. Two overlapping ellipses

As another example, consider now the ellipses defined by $eleq_i(1.25,0.75,-1,-2,\pi/3)(x,y)$ and $eleq_j(1.5,0.83,-0.5,1,\pi/4)(x,y)$. The non-overlapping value between the ellipses is 1.84 with $(x,y) = (-0.758,-0.141)$: again, this can be found by solving equation (10). Figure 3 shows this non-overlapping configuration.

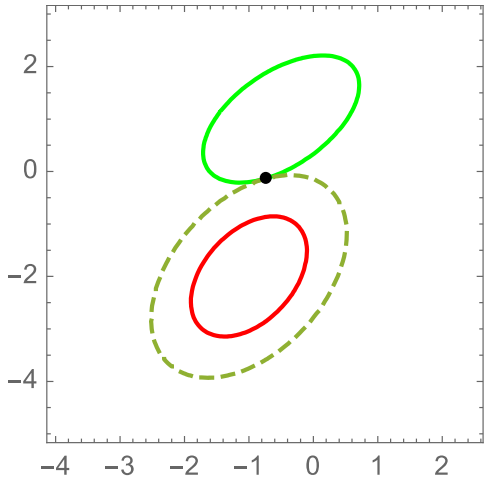

Figure 3. Two non-overlapping ellipses

In the overall optimization strategy, λ_i are the Lagrange multpliers in the equations for finding the point (xm_i, ym_i) on ellipse i that is most distant from the origin. The calculation is restricted to maximization by restricting the sign of λ_i to be postive. The square of the radius of the circumscribing circle satisfies the relation $rc^2 \geq xm_i^2 + ym_i^2$ for all i. In addition, $\lambda_{i,j}$ are the Lagrange multpliers in the equations for finding the point $(x_{j,i}, y_{j,i})$ on ellipse j that minimizes the value of the equation describing ellipse i. This calculation is restricted to minimization by requiring the value of $\lambda_{i,j}$ to be negative. Finally, there are the constraints that prevent ellipse i from overlapping with ellipse j, by requiring the value of the equation describing ellipse i on the point on ellipse j that minimizes that value to be at least ε. Values of $\varepsilon = 0.00$, 1E-6, and 0.01 have been used successfully: hence, the model and the optimization solver are not too sensitive to the choice of this parameter.

To summarize the model development steps described above, we obtain the following optimization model for the case of n ellipses.

$$\text{minimize} \qquad rc \tag{11}$$

$$\text{subject to} \qquad rc^2 \geq xm_i^2 + ym_i^2 \qquad \text{for } i = 1, \ldots, n$$

$$xm_i^2 + ym_i^2 \geq \min(a_i, b_i)^2 \qquad \text{for } i = 1, \ldots, n$$

$$2 \cdot xm_i = \lambda_i \cdot eleqdx(a_i, b_i, xc_i, yc_i, \theta_i)(xm_i, ym_i) \qquad \text{for } i = 1, \ldots, n$$

$$2 \cdot ym_i = \lambda_i \cdot eleqdy(a_i, b_i, xc_i, yc_i, \theta_i)(xm_i, ym_i) \qquad \text{for } i = 1, \ldots, n$$

$$eleq(a_i, b_i, xc_i, yc_i, \theta_i)(xm_i, ym_i) = 0 \qquad \text{for } i = 1, \ldots, n$$

$$eleqdx(a_i, b_i, xc_i, yc_i, \theta_i)(x_{j,i}, y_{j,i}) = \lambda_{j,i} \cdot eleqdx(a_j, b_j, xc_j, yc_j, \theta_j)(x_{j,i}, y_{j,i}) \quad \text{for } i = 1, \dots, n-1,\ j = i+1, \dots, n$$

$$eleqdy(a_i, b_i, xc_i, yc_i, \theta_i)(x_{j,i}, y_{j,i}) = \lambda_{j,i} \cdot eleqdy(a_j, b_j, xc_j, yc_j, \theta_j)(x_{j,i}, y_{j,i}) \quad \text{for } i = 1, \dots, n-1,\ j = i+1, \dots, n$$

$$eleq(a_j, b_j, xc_j, yc_j, \theta_j)(x_{j,i}, y_{j,i}) = 0 \quad \text{for } i = 1, \dots, n-1,\ j = i+1, \dots, n$$

$$eleq(a_i, b_i, xc_i, yc_i, \theta_i)(x_{j,i}, y_{j,i}) \geq \varepsilon \quad \text{for } i = 1, \dots, n-1,\ j = i+1, \dots, n$$

$$lb \leq xc_i \leq ub \quad \text{for } i = 1, \dots, n$$

$$lb \leq yc_i \leq ub \quad \text{for } i = 1, \dots, n$$

$$-\pi \leq \theta_i \leq \pi \quad \text{for } i = 1, \dots, n$$

$$lb \leq xm_i \leq ub \quad \text{for } i = 1, \dots, n$$

$$lb \leq ym_i \leq ub \quad \text{for } i = 1, \dots, n$$

$$lb \leq x_{j,i} \leq ub \quad \text{for } i = 1, \dots, n-1,\ j = i+1, \dots, n$$

$$lb \leq y_{j,i} \leq ub \quad \text{for } i = 1, \dots, n-1,\ j = i+1, \dots, n$$

$$0 \leq \lambda_i \leq 2 \cdot ub \quad \text{for } i = 1, \dots, n$$

$$2 \cdot lb \leq \lambda_{j,i} \leq 0 \quad \text{for } i = 1, \dots, n-1,\ j = i+1, \dots, n$$

Here lb and ub are lower and upper bounds defined for each ellipse packing instance in order to facilitate achieving feasible solutions.

The optimization model (11) has $1 + 6n + (n-1)^2$ decision variables and, in addition to the bound constraints that are imposed on all decision variables, $5n + 4(n-1)^2$ nonlinear constraints: the latter constraints are all non-convex.

Considering the formulas introduced earlier for the ellipses, model (11) represents a highly nonlinear (global) optimization problem-class in which both the number of variables and constraints increases quadratically as a function of n. As an example, to solve a packing problem with $n = 10$ ellipses, we have a model-instance with 142 decision variables and corresponding bound constraints, and 374 non-convex (nonlinear) constraints.

Based on these observations, we conjecture that the general computational difficulty of model (11) will rapidly increase as a function of the number of packed ellipses n.

3 Numerical Global Optimization for Packing Ellipses

3.1 Global Optimization: Basic Concepts

The objective of global optimization (GO) is to find the "absolutely best" solution of provably or potentially multi-extremal problems. Most object packing problems are provably multi-modal, often possessing a large number of local optima. Without going into technical details, a simple inspection of the relations leading to the problem statement (11) implies that general ellipse packings belong to a difficult GO model category.

As noted earlier, one cannot expect to find analytical solutions to general object packing problems – even when considering far less complicated model types than the one studied here. Therefore we have been applying global-local numerical optimization to handle various object configuration (such as spherical point and circle packing) problems, to produce high quality feasible solutions to non-trivial model instances: for details, cf. e.g. Pintér (2001), Stortelder *et al.* (2001), Pintér and Kampas (2005, 2006), Kampas and Pintér (2006), Castillo *et al.* (2008), Pintér and Kampas (2013).

In our present study, we apply the Lipschitz Global Optimizer (LGO) solver system for global-local nonlinear optimization, in its implementation linked to the computing system *Mathematica*. First, we review some basic technical requirements related to using LGO: this is followed by a concise discussion of the key LGO features.

The objective of GO is expressed mathematically as follows. We seek for the best decision expressed by a real n-vector $x \in R^n$ that satisfies a set of feasibility constraints, and minimizes (or maximizes) the value of a given objective function. A corresponding high-level optimization model statement is given by

$$\text{minimize } f(x) \text{ subject to } x \in D. \tag{12}$$

Here f denotes the objective function of the decision problem, and D denotes the set of feasible solutions.

Since the analytical solution of GO problems is not possible for a very large variety of model instances, our practical goal is to solve instances of problem (12) numerically. To this end, we shall assume that $D \subset R^n$ is a bounded robust set (i.e., D is the closure of a non-empty open set) and that f is a continuous function. Under these basic conditions – by the classical theorem of Weierstrass – the set of global solutions to (12) is non-empty. We denote this set by X^*, and remark that in many cases of practical relevance, X^* consists of a single point x^*, or possibly of several isolated points. For reasons of

algorithmic tractability, it will be assumed that X^* is at most countable. Introducing the notation $f^* = f(x^*)$, to solve (12) theoretically means the following requirement:

"find all elements of X^*, and the function value f^*." (13)

In numerous cases, it can become very difficult – or even impossible − to solve (12) in the exact sense of (13). Consequently, often numerical approximations of (13) need to be used. A frequently stated approximate solution requirement is to find a feasible point x which yields an objective function value that is "sufficiently close" to f^*. Hence, our goal can be described as follows:

"find an $x \in D$ such that $f(x) \leq f^* + \delta$." (14)

In (14), $\delta > 0$ is a tolerance parameter.

To guarantee the numerical solvability of model (12) in the sense of (14) – on the basis of a finite set of algorithmically generated search (sample) points from D – we need to postulate also some quantified analytical property of f that is valid over D. As an important example, the Lipschitz-continuity of f is frequently postulated, when appropriate. That is, for all point pairs x_1 and x_2 from D, we assume the relation

$$|f(x_1) - f(x_2)| \leq L\big||x_1 - x_2|\big|. \tag{15}$$

In (15), $L = L(f, D)$ is a suitable Lipschitz constant of f that is valid over the set D. Inequality (15) guarantees that the possible "variability" of function f is uniformly controlled by the respective changes of its argument. The Lipschitz assumption is met, for instance, by all continuously differentiable functions f defined over the compact set D. However, the best (minimal) value of L is typically unknown and to find it would require the numerical solution of another GO problem: hence, in practice suitable values of L typically need to be estimated based on the generated search points.

In many practical applications of optimization, D is defined by explicit, finite lower and upper bound vectors lb and ub regarding x, and by a finite number of additional continuous or Lipschitz-continuous constraints; that is,

$$D = \{lb \leq x \leq ub, g_i(x) \leq 0 \text{ for } i = 1, \ldots, m\}. \tag{16}$$

Obviously, in (16) all bound constraints and inequality constraints are to be interpreted component-wise (while in model (11) we considered bound and general constraints individually, in order to provide a detailed description). To maintain a simple standardized notation, all general inequality constraints are considered in the form $g_i(x) \leq 0$ as shown by (16). (All inequality constraints can be simply (re)written in this form, and equality constraints can be replaced by a pair of inequality constraints.)

3.2 The LGO Solver System for Global-Local Nonlinear Optimization

Within the general modeling framework outlined in Section 3.1, the LGO software package is aimed at finding the numerical global optimum of model instances from a very general class of continuous global optimization problems. LGO has been in use since the early 1990s, and it has been documented in detail by other publications and technical reports. In particular, Pintér (1996) presents a theoretical exposition of adaptive deterministic partition strategies and stochastic search methods to solve global optimization problems under continuity or Lipschitz-continuity assumptions. The exhaustive search capability of such algorithmic procedures guarantees their theoretical global convergence. Various implementation aspects and a range of application areas with detailed case studies are also discussed in the book.

The core solver system in LGO with implementations for various modeling platforms has been described by Pintér (1996, 1997, 2002, 2005, 2007, 2009), Pintér *et al*. (2006). For more recent development work including benchmarking studies, consult e.g. Çağlayan and Pintér (2013), Pintér and Horváth (2013), Pintér and Kampas (2013), Pintér (2014). Further technical details are discussed by the current LGO manual (Pintér, 2016), which includes a fairly extensive list of topical references. Therefore here we present only a very brief summary of key LGO features and implementation details pertinent to this work.

The core (Fortran or C/C++/C# compiler platform based) LGO solver suite seamlessly integrates several derivative-free global and local optimization strategies, without requiring higher-order (gradient or Hessian) information. The strategies referred to include regularly spaced sampling, as a global presolver (RSS); a branch-and-bound global search method (BB); global adaptive random search (GARS); Multi-start based global random search (MS); and local search (LS). According to extensive numerical experience, in complicated GO models, MS (with added LS solver phases) often finds the best numerical solution. For this reason, MS is the recommended default LGO solver option that has been used also in our present numerical study.

3.3 *MathOptimizer Professional*

LGO has been made available for a number of model development platforms as a (commercial) solver option: these platforms currently include AMPL, GAMS, MPL, Excel, Maple, *Mathematica* and Matlab. Similarly to our earlier circle packing studies, the ellipse packing model has been implemented in *Mathematica* (Wolfram Research, 2015): therefore we use here the LGO implementation linked to *Mathematica.* This implementation, with the software product name *MathOptimizer Professional*, has been extensively used also in our benchmarking studies: cf. e.g. Pintér and Kampas (2013). Again, we only summarize the key features of this software that are relevant for the present discussion, and refer for further details to the works cited earlier, as well as to the current *MathOptimizer Professional* documentation (Pintér and Kampas, 2015). This user's guide is a "live" (fully interactive) *Mathematica* notebook document with readily executable examples.

The *MathOptimizer Professional* software package combines *Mathematica*'s powerful optimization model development capabilities with the external LGO solver suite. To emphasize this point, let us remark that the entire ellipse packing model and its solution consist "only" of about 50 carefully crafted *Mathematica* code lines including the code for displaying the ellipse configurations found. *MathOptimizer Professional* automatically transforms optimization models formulated in *Mathematica* into C or Fortran code (whenever this is possible): this translated model is handed over directly to LGO for solution. Following a seamless optimization model compilation, linking, and execution procedure, the optimization results are directly returned to the calling *Mathematica* document. This approach can lead to a significant program execution speedup when compared to native optimization in *Mathematica*: the speedup becomes increasingly more noticeable for larger models. We also note that the core LGO solver performance compares favorably to the corresponding solver features of *Mathematica. MathOptimizer Professional* can be used to handle sizeable models, currently with thousands of variables and general constraints.

4 Illustrative Numerical Results

To our best knowledge, there are no previously studied model instances available for the general ellipse packing problem considered in our present work. The problem instances summarized in Table 1 are taken from Kallrath and Rebennack (2014) – recalling that their work was aimed at packing ellipses in optimized rectangles. This choice of tests instances allows comparisons regarding the packing density of rectangular vs. circular packings (not in a competitive sense, since the configuration geometries are different).

Our calculations were performed on a PC with a quad-core Intel i7 processor running at 3.7 GHz, with 16 GBytes of RAM, using *MathOptimizer Professional* running in *Mathematica* version 10, and using the GCC compiler to generate the files for LGO.

Table 1. Ellipse packing instances

Test case	(a_i, b_i)	Total area to be packed
ax2a	(2.0,1.5), (1.5,1.0)	14.13717
ax2b	(2.0,1.5), (1.8,1.4)	17.34159
ax3a	ax2a + (1.0,0.8)	16.65044
ax3b	ax2b + (0.8,0.7)	19.10088
ax4a	ax3a + (0.9,0.75)	18.77102
ax4b	ax3b + (1.1,1.0)	22.55664
ax5a	ax4a + (0.8,0.6)	20.27898
ax5b	ax4b + (0.9,0.8)	24.81858
ax6	ax5a + (0.7,0.3)	20.93872
ax11	(2.0,1.5), (1.8,1.5), (1.6,1.5), (1.5,1.2), (1.3,1.0), (1.2,0.9), (1.1,0.8), (1.0,0.75), (0.9,0.6), (0.8,0.5),(0.7,0.3)	47.31239
ax14	$7 \cdot (1.0,0.75) + 7 \cdot (0.5,0.375)$	20.6167

Table 2 summarizes the computational results, noting that CPU times are reasonable even for the last two (largest) problem instances. Note that Table 2 also includes the packing fraction and the maximum constraint violation. In Table 3, we summarize our results for general ellipse packings in an optimized circle next to the best solutions found for packings in an optimized rectangle given by Kallrath and Rebennack (2014). As mentioned, the configuration geometries are rather different given the different optimized containers; but we still get an overall impression regarding the sort of packing densities that can be achieved for rectangles and circles, for a range of model instances.

Table 2. Ellipse packing results

Test case	Packing radius rc	Area of optimized container	Packing fraction	Time (sec)	Max constraint violation
ax2a	2.49873	19.61501	0.72073	0.5	8E-9
ax2b	2.9	26.42079	0.65636	0.6	1E-9
ax3a	2.56257	20.63010	0.80709	1.0	5E-10
ax3b	2.9	26.42079	0.72295	1.0	5E-10
ax4a	2.74972	23.75346	0.79024	3.1	4E-9
ax4b	2.98985	28.08333	0.80320	3.0	5E-9
ax5a	2.84911	25.50165	0.79520	7.6	3E-12
ax5b	3.26085	33.40500	0.74296	7.7	9E-9
ax6	2.89647	26.35651	0.79444	20.0	4E-9
ax11	4.35292	59.52662	0.79481	31.0	5E-10
ax14	2.864	25.76890	0.80006	106.0	1E-7

Table 3. Packing results in a circle and packing results in a rectangle

Test case	Circular container		Rectangular container	
	Area of optimized container	Packing fraction	Area of optimized container	Packing fraction
ax2a	19.61501	0.72073	18.00000	0.78540
ax2b	26.42079	0.65636	22.23152	0.78005
ax3a	20.63010	0.80709	21.38577	0.77858
ax3b	26.42079	0.72295	25.22467	0.75723
ax4a	23.75346	0.79024	23.18708	0.80955
ax4b	28.08333	0.80320	28.54159	0.79031
ax5a	25.50165	0.79520	25.29557	0.80168
ax5b	33.40500	0.74296	31.28873	0.79321
ax6	26.35651	0.79444	25.51043	0.82079
ax11	59.52662	0.79481	64.59177	0.73248
ax14	25.76890	0.80006	29.65886	0.69513

Illustrative packing configurations for the last two (largest) problem instances ax11 and ax14 are given in Figures 5 and 6, respectively. The points shown on the ellipses and optimizer container are the subsidiary points for preventing ellipse overlap and

determining the radius of the circumscribing circle. Note that for preventing the overlap of a pair of ellipses, there is only a point on one of the two ellipses, not on both.

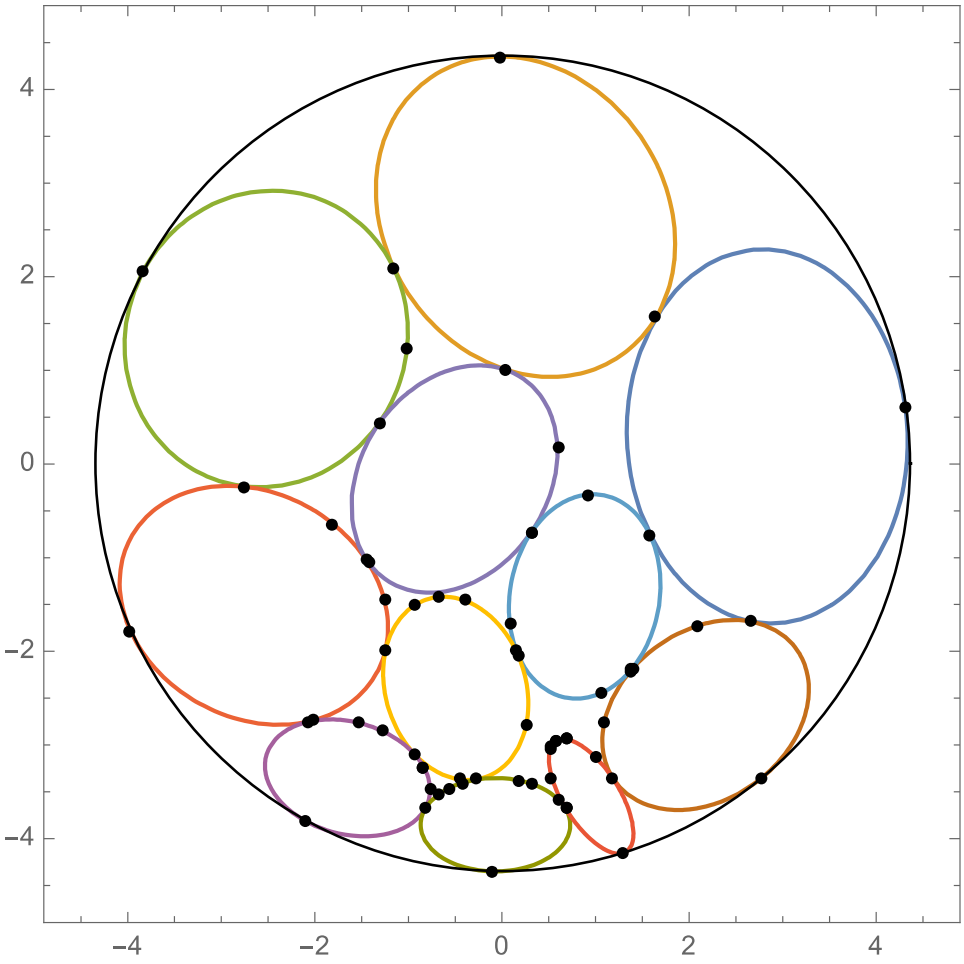

Figure 5. Packing for configuration ax11

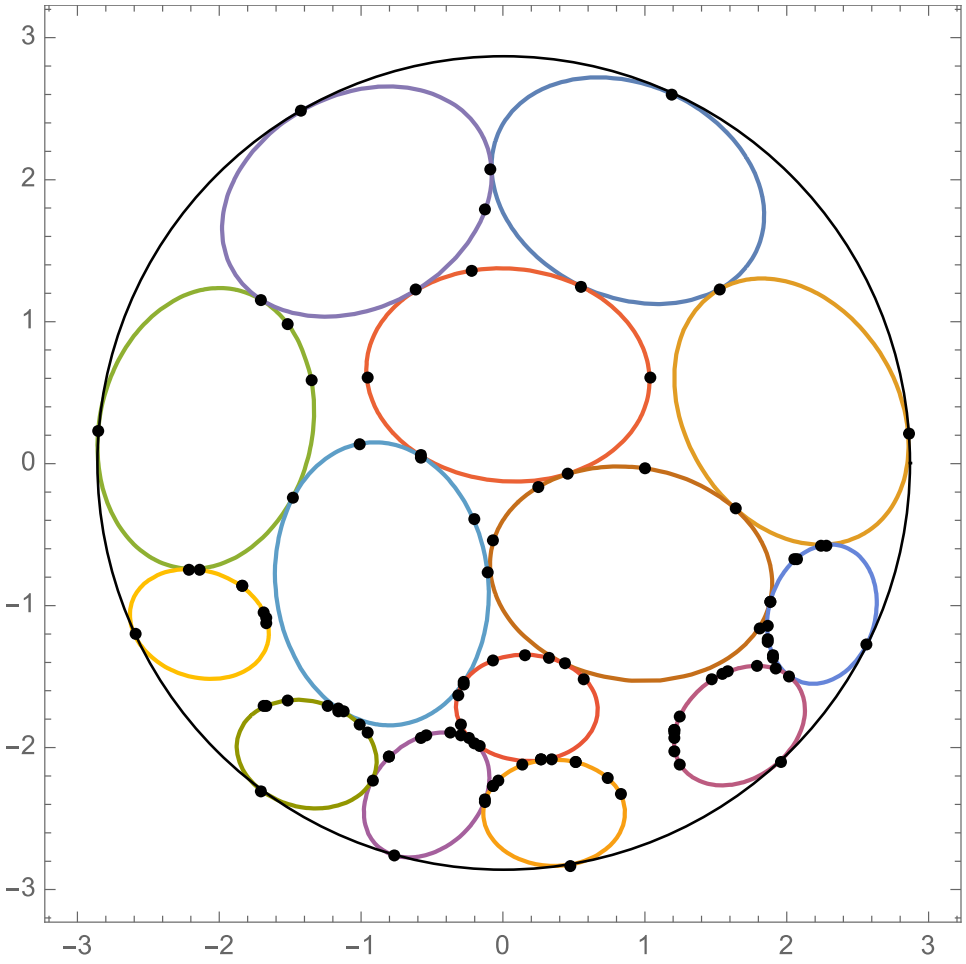

Figure 6. Packing for configuration ax14

5 Summary and Conclusions

In this study, we considered the general ellipse packing problem with respect to an optimized circle container with minimal radius. Our review of the related literature, even if not all works cited here aim at handling the exact same problem-type, illustrates the significant difficulty of similar packing problems.

One of our key contributions is the modeling and enforcement of constraints that fall into two groups. The first group represents the constraints that keep the ellipses inside the circumscribing circle and prevent them from overlapping. (Recall that preventing ellipse overlap depends on both the orientation of the ellipses and the location of their centers.) The second group represents the equations generated by the embedded Lagrange multiplier conditions. In our global optimization strategy, preventing ellipse overlaps proceeds simultaneously with the minimization of the radius of the circumscribing circle. We applied the LGO solver system for global-local nonlinear optimization (with the software product name *MathOptimizer Professional*) in an implementation linked to the computing system *Mathematica*. Our results demonstrate that the embedded Lagrangian

multipliers based modeling approach combined with global optimization enables the computational solution of difficult ellipse packing problems with several hundred variables and general non-convex constraints. Preliminary further work indicates that our new model formulation approach has the potential to extend to ellipse packing problems in other types of container sets.

References

Addis, B., Locatelli, M., Schoen, F., 2008. Efficiently packing unequal disks in a circle. *Operations Research Letters* 36, 37-42.

Birgin, E.G., Bustamante, L.H., Flores Callisaya, H., Martínez, J.M., 2013. Packing circles within ellipses. *International Transactions in Operational Research* 20, 365-389.

Çağlayan, M.O., Pintér, J.D., 2013. Development and calibration of a currency trading strategy using global optimization. *Journal of Global Optimization* 56, 353–371.

Castillo, I., Kampas, F.J., Pintér, J.D., 2008. Solving circle packing problems by global optimization: Numerical results and industrial applications. *European Journal of Operational Research* 191, 786-802.

Castillo, I., Sim, T., 2004. A spring-embedding approach for the facility layout problem. *Journal of the Operational Research Society* 55, 73-81.

Fasano, G. Pintér, J.D. (Eds.), 2015. *Optimized Packings with Applications*. Springer Science + Business Media, New York.

Galiev, S.I., Lisafina, M.S., 2013. Numerical optimization methods for packing equal orthogonally oriented ellipses in a rectangular domain. *Computational Mathematics and Mathematical Physics* 53, 1748-1762.

Gensane, T., Honvault, P., 2014. Optimal packings of two ellipses in a square. *Forum Geometricorum* 14, 371-380.

Grosso, A., Jamali, A.R.M.J.U., Locatelli, M., Schoen, F., 2010. Solving the problem of packing equal and unequal circles in a circular container. *Journal of Global Optimization* 47, 63-81.

Hifi, M., M'Hallah, R., 2009. A literature review on circle and sphere packing problems: models and methodologies. *Advances in Operations Research* 2009, Article ID 150624, 22 pages, doi:10.1155/2009/150624.

Kallrath, J., Rebennack, S., 2014. Cutting ellipses from area-minimizing rectangles. *Journal of Global Optimization* 59, 405-437.

Kampas, F.J., Pintér, J.D., 2006. Configuration analysis and design by using optimization tools in *Mathematica*. *The Mathematica Journal* 10, 128-154.

Litvinchev, I., Infante, L., Ozuna, L., 2015. Packing circular-like objects in a rectangular container. *Journal of Computer and Systems Sciences International* 54, 259-267.

Markót, M.Cs., 2005. Optimal packing of 28 equal circles in a unit square – the first reliable solution. *Numerical Algorithms* 37, 253-261.

Pintér, J.D., 1996. *Global Optimization in Action.* Kluwer Academic Publishers, Dordrecht. (Now distributed by Springer Science + Business Media, New York.)

Pintér, J.D., 1997. LGO − A program system for continuous and Lipschitz global optimization. In: Bomze, I., Csendes, T., Horst, R., Pardalos, P.M. (Eds.). *Developments in Global Optimization*, pp. 183-197. Kluwer Academic Publishers, Dordrecht.

Pintér, J.D., 2001. Globally optimized spherical point arrangements: model variants and illustrative results. *Annals of Operations Research* 104, 213-230.

Pintér, J.D., 2002. Global optimization: software, test problems, and applications. In: Pardalos, P.M., Romeijn, H.E. (Eds.). *Handbook of Global Optimization*, Vol. 2., pp. 515-569. Kluwer Academic Publishers, Dordrecht.

Pintér, J.D., 2005. Nonlinear optimization in modeling environments: software implementations for compilers, spreadsheets, modeling languages, and integrated computing systems. In: Jeyakumar, V., Rubinov, A.M., Eds. *Continuous Optimization: Current Trends and Applications*, pp. 147-173. Springer Science + Business Media, New York.

Pintér, J.D., 2007. Nonlinear optimization with GAMS/LGO. *Journal of Global Optimization* 38, 79–101.

Pintér, J.D., 2009. Software development for global optimization. In: Pardalos, P.M. and T. F. Coleman, Eds. *Global Optimization: Methods and Applications*, pp. 183-204. *Fields Institute Communications Volume 55*. Published by the American Mathematical Society, Providence, RI.

Pintér, J.D., 2014. How difficult is nonlinear optimization? A practical solver tuning approach, with illustrative results. (Submitted for publication) Available at http://www.optimization-online.org/DB_HTML/2014/06/4409.html.

Pintér, J.D., 2016. *LGO – A Model Development and Solver System for Global-Local Nonlinear Optimization, User's Guide.* (Current edition) Published and distributed by Pintér Consulting Services, Inc., Halifax, NS, Canada.

Pintér, J.D., Kampas, F.J., 2005. Nonlinear optimization in *Mathematica* with *MathOptimizer Professional*. *Mathematica in Education and Research* 10, 1-18.

Pintér, J.D., Kampas, F.J., 2006. *MathOptimizer Professional*: key features and illustrative applications. In: Liberti, L., Maculan, N. (Eds.), *Global Optimization: From Theory to Implementation*. Springer Science + Business Media, New York, 263-280.Pintér, J.D., Linder, D. and Chin, P., 2006. Global Optimization Toolbox for Maple: An introduction with illustrative applications. *Optimization Methods and* Software 21, 565-582.

Pintér, J.D., Horváth, Z., 2013. Integrated experimental design and nonlinear optimization to handle computationally expensive models under resource constraints. *Journal of Global Optimization* 57, 191-215.

Pintér, J.D., Kampas, F.J., 2013. Benchmarking nonlinear optimization software in technical computing environments. I. Global optimization in *Mathematica* with *MathOptimizer Professional*. *TOP* 21, 133-162.

Pintér, J.D., Kampas, F.J., 2015. *Getting Started with MathOptimizer Professional*. Published and distributed by Pintér Consulting Services, Inc., Halifax, NS, Canada.

Riskin, M.D., Bessette, K.C., Castillo, I., 2003. A logarithmic barrier approach to solving the dashboard planning problem. *INFOR* 41, 245-257.

Stortelder, W.J.H., de Swart, J.J.B., Pintér, J.D., 2001. Finding elliptic Fekete point sets: Two numerical solution approaches. *Journal of Computational and Applied Mathematics* 130, 205-216.

Szabó, P.G., Csendes, T., Casado, L.G., García, I., 2001. Equal circles packing in a square I – Problem setting and bounds for optimal solutions. In: Giannessi, F., Pardalos, P.M., Rapcsák, T. (Eds.), *Optimization Theory: Recent Developments from Mátraháza*. Kluwer, Dordrecht.

Szabó, P.G., Markót, M.C., Csendes, T., 2005. Global optimization in geometry – circle packing into the square. In: Audet, P., Hansen, P., Savard, G. (Eds.), *Essays and Surveys in Global Optimization.* Kluwer, Dordrecht.

Szabó, P.G., Markót, M.C., Csendes, T., Specht, E., Casado, L.G., García, I., 2007. *New Approaches to Circle Packing in a Square with Program Codes.* Springer Science + Business Media, New York.

Uhler, C., Wright, S.J., 2013. Packing ellipsoids with overlap. *SIAM Review* 55, 671-706.

Wolfram Research, 2015. *Mathematica* (Current release 10.3, December 2015). Wolfram Research, Inc., Champaign, IL.