

Optimized Ellipse Packings in Regular Polygons

Frank J. Kampas ^a, Ignacio Castillo ^{b,*}, János D. Pintér ^c

^a Physicist at Large Consulting LLC, Bryn Mawr, PA, USA

^b Lazaridis School of Business and Economics, Wilfrid Laurier University, Waterloo, ON, Canada

^c Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

* Corresponding author. E-mail address: icastillo@wlu.ca

Abstract

We present model development and numerical solution approaches to the problem of packing a general set of ellipses without overlaps into an optimized polygon. Specifically, for a given set of ellipses, and a chosen integer $m \geq 3$, we minimize the apothem of the regular m -polygon container. Our modeling and solution strategy is based on the concept of embedded Lagrange multipliers. To solve models with up to $n \leq 10$ ellipses, we use the LGO solver suite for global-local nonlinear optimization. In order to reduce increasing runtimes, for model instances with $10 \leq n \leq 20$ ellipses, we apply local search launching the Ipopt solver from selected random starting points. The numerical results demonstrate the applicability of our modeling and optimization approach to a broad class of highly non-convex ellipse packing problems, by consistently returning good quality feasible solutions in all (231) illustrative model instances considered here.

Key words: General Ellipse Packings in Regular Polygons · Model Development Using Embedded Lagrange Multipliers · Global and Local Nonlinear Optimization · LGO Solver Suite · Random Starting Points and Local Search by Ipopt · Numerical Results.

1 Introduction and Motivation

Object packing problems (OPPs) arise in relation to a broad range of engineering and scientific applications. First, we present a concise review of some well-frequented as well as several less known OPPs, comment on solution approaches, and highlight various applications.

A finite circle packing is an optimized non-overlapping arrangement of n circles inside a container set such as a circle, square or a general rectangle. This problem – in particular, the case of packing identical circles – has received considerable attention in the literature. The object-symmetric structure of identical circle packings (ICPs) makes these problems easier, but finding optimal configurations for arbitrary values of n is still a difficult challenge. Studies of packing identical circles frequently aim at proving the optimality of the configurations found, either theoretically or with the help of rigorous computational approaches. As of today, *provably optimal* configurations are known only for tens of circles, with the exception of certain special cases; *best known numerical results* are available for packing up to 2,600 circles in a circle, and 10,000 circles in a square. For further details and references, consult e.g., Szabó *et al.* [51] or Specht [47].

The general circle packing (GCP) problem defined for collections of n (in principle) arbitrary sized circles is a substantial generalization of the ICP case. Provably optimal GCP configurations can be found only to models with $n \leq 4$. Therefore GCP studies frequently utilize generic or tailored global scope numerical solution strategies. Without going into further details related to circle packings, we refer to Castillo *et al.* [7] and to Hifi and M'Hallah [16] for reviews of both ICPs and GCPs and their applications, noting that Pintér *et al.* [41] present numerical results for generalized d -sphere packings in $d = 2, 3, 4, 5$ dimensions with up to 50 spheres.

Compared to ICPs and GCPs, ellipse packing problems (EPPs) have received relatively little attention in the literature so far. Finding globally optimized packings of ellipses with arbitrary size and orientation is a hard computational problem. The key challenge is the modeling and enforcement of the constraints to avoid ellipse overlaps, as a function of ellipse center locations and orientations.

To illustrate the difficulty of this class of problem, first we mention an exact result that deals with the densest packing of just *two* non-overlapping *congruent* ellipses in a square. For this very special case, Gensane and Honvault [14] analytically define the densest packing of two identical ellipses with aspect ratio r , for all real numbers r in $[0, 1]$. Honvault [18] analytically describes the densest packing of three non-overlapping congruent ellipses in a square.

Galiev and Lisafina [13] study the problem of packing *identically sized* and *orthogonally oriented* ellipses inside a rectangular container. Binary linear optimization models are proposed using a grid that approximates the container region, and then considering the nodes of the grid as potential positions for the ellipse centers. Two special cases regarding the orientation of the ellipses are considered: i) the major axes of all 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 for all others they are parallel to the y axis. A heuristic algorithm based on binary linear model formulations is proposed, with numerical results.

Litvinchev *et al.* [26] investigate optimized packings of so-called *circular-like* objects – including circles, ellipses, rhombuses, and octagons – in a rectangular container. Similarly to Galiev and Lisafina [13], they propose a binary linear optimization model formulation based on a grid that approximates the container. The resulting problem is then solved using the software package CPLEX. Numerical results related to packing identically sized and oriented circles, ellipses, rhombuses, and octagons are presented.

Uhler and Wright [54] study the problem of packing arbitrary size 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 illustrate their approach applied to modeling chromosome organization in the human cell nucleus.

Kallrath and Rebennack [22] address the problem of packing ellipses of arbitrary size and orientation into an optimized rectangle. The packing model formulation is introduced as a cutting problem. Their key idea is to use separating lines to ensure that the ellipses do not overlap with each other, using hyperplanes and coordinate transformations. For problem instances with $n \leq 14$ ellipses, the authors find globally optimal numerical solutions (considering the finite arithmetic precision of the global solvers used). At the same time, for their $n > 14$ ellipse-based model instances none of the (local or global) nonlinear

optimization solvers tried by them in the GAMS modeling environment could compute even a feasible solution. Therefore, they propose heuristic approaches in which ellipses are added sequentially to find an approximately optimized rectangular container: this strategy allows the computation of high quality solutions for up to 100 ellipses. Kallrath [21] extends the work presented in Kallrath and Rebennack [22], to pack ellipsoids into optimized rectangular boxes.

Birgin *et al.* [3, 4] propose several model formulations for packing ellipsoids in a container. Specifically, they address various two- and three-dimensional EPPs with rectangular containers (for both identical and arbitrary sized ellipses), elliptical containers (for identical ellipses), spherical containers (for identical ellipsoids), and cuboid containers (for identical ellipsoids). The authors propose multi-start global scope optimization procedures that use starting guesses, followed by using local optimization solvers, in order to find good quality solutions with up to 1000 ellipsoids.

Stoyan *et al.* [49] further develop their phi-function technique (cf. Stoyan *et al.* [50]) to pack ellipses into rectangular containers of minimal area. These functions – referred to as quasi-phi-functions – support the packing of non-overlapping ellipses with arbitrary size and orientation. The authors develop an efficient solution algorithm based on local optimization combined with a feasible region transformation procedure. This procedure reduces the dimension of the problem instance, which allows the authors to use a local solver (Ipopt in their case) to find a high quality solution. Stoyan *et al.* [49] present computational results that compare favorably with those presented by Kallrath and Rebennack [22].

In addition to being an interesting model development and optimization challenge *per se*, ellipse and ellipsoid packings have important scientific and industrial applications. To illustrate this aspect, we mention studies related to the structure of liquids, crystals, and glasses (Bernal [2]); the flow and compression of granular materials (Edwards [12], Jaeger and Nagel [19], Jaeger *et al.* [20]); the design of high density ceramic materials, and the formation and growth of crystals (Cheng *et al.*, [10], Rintoul and Torquato [44]); the thermodynamics of liquid to crystal transition (Alder and Wainwright [1], Chaikin [9], Pusey [43]); and chromosome organization in human cell nuclei (Uhler and Wright [54]).

In this article, we study the packing of ellipses with arbitrary size and orientation in regular m -polygons: our objective is to minimize the area of the container polygon. Packing ellipses into a regular polygon requires i) the determination of the maximal distance from the center of all polygon faces 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 length of the polygon’s apothem (the line segment from the center of the polygon to the midpoint of one of its sides, see Figure 1), which is then to be minimized. The second requirement serves to prevent the ellipses from overlapping. Explicit analytical formulas for the first requirement can be directly derived, and used in our optimization strategy. However, for the second requirement explicit analytical formulas – even if they exist – would be complicated. Therefore, our modeling and solution approach is based on embedding optimization calculations, using Lagrange multipliers, into the overall optimization strategy. In this Lagrangian framework, optimization proceeds simultaneously considering both requirements.

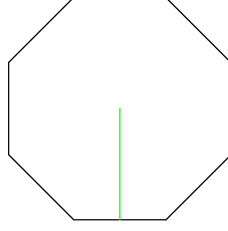


Figure 1. Apothem of a regular polygon ($m = 8$)

To summarize, we propose a modeling and optimization framework that uses embedded Lagrange multipliers to prevent the ellipses from overlapping. This allows us to solve the container minimization problem numerically with a single call to a suitable optimization procedure. Our model formulation for regular polygons allows the high quality approximations of a range of container shapes. This is particularly important in so called “open-field” or “green-field” situations where a set of regular polygonal containers can adequately approximate the field under consideration. Our model can also be extended to consider general (not necessarily regular) convex polygons: see Giachetti and Sanchez [15] for an interesting design application.

Our modeling and optimization strategy consistently provides high quality feasible solutions to non-trivial model instances. We are able to reproduce all previous related results considered here, improving some of the previously reported best numerical results. We also solve new problem instances for a range of model parameter choices.

2 Optimization Model Development

Our objective is to minimize the area of the regular m -polygon that contains, without overlaps, a given collection of n ellipses with arbitrary size and shape characteristics. The input data to these optimization problem instances are m , n , and the (semi-major and semi-minor) axes (a_i, b_i) of the ellipses to be packed for $i = 1, \dots, n$. The primary decision variables are the polygon’s apothem d , and the centre position (x_i^c, y_i^c) and orientation θ_i of the packed ellipses, $i = 1, \dots, n$. Secondary (induced) variables are the positions of the distance maximizing lines pointing from each ellipse boundary to the center of each of the polygon faces, 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. All listed secondary variables are implicitly determined by the primary variables, as discussed below.

The model constraints belong to two groups. The first constraint group uses the secondary variables to keep the ellipses inside the container, and to prevent them from overlapping. The second constraint group represents the equations generated by the embedded Lagrange multiplier conditions. The calculations for optimizing the polygon and for preventing ellipse overlaps proceed simultaneously, rather than being performed to completion at each optimization step. Observe that since the area of the polygon equals $m \cdot d^2 \cdot \tan(\pi/m)$, m being an input parameter, finding the minimal apothem d is equivalent to minimizing the area of the regular polygon.

Omitting first the index i (for a simpler notation) equation $e(a, b, x^c, y^c, \theta; x, y) = 0$ introduced by (1) defines the boundary of an ellipse with semi-major and semi-minor axes a and b , centered at $\{x^c, y^c\}$, and rotated counterclockwise by angle θ . The formula for $e(a, b, x^c, y^c, \theta; x, y)$ is obtained by transforming the equation of a circle with radius 1, centered at $(0,0)$, as shown below.

$$\begin{aligned} e(a, b, x^c, y^c, \theta; x, y) &= \left(\frac{\cos(\theta)(x - x^c)}{a} + \frac{\sin(\theta)(y - y^c)}{a} \right)^2 \\ &+ \left(\frac{\cos(\theta)(y - y^c)}{b} - \frac{\sin(\theta)(x - x^c)}{b} \right)^2 - 1 = 0. \end{aligned} \quad (1)$$

In (1) the coordinate system is rotated clockwise by an angle of $-\theta$, which is equivalent to rotating the ellipse counterclockwise by θ around its centre. The value of $e(a, b, x^c, y^c, \theta; x, y)$ is negative for all points (x, y) located inside the ellipse, zero for all points on the ellipse boundary, and positive for all points outside the ellipse. Let us recall here that (x^c, y^c) and θ are primary decision variables for each ellipse i : in general, these variables will be denoted by (x_i^c, y_i^c) and θ_i for $i = 1, \dots, n$.

In order to standardize our model formulation, we assume that the optimized regular polygon container is centered at the origin. Consider now the line $\gamma_x x + \gamma_y y = \ell$ that embeds one of the polygon sides. The slope of this line is $-\gamma_x/\gamma_y$, and the distance from the origin to the line equals

$$\sqrt{\frac{\ell^2}{\gamma_x^2 + \gamma_y^2}}. \quad (2)$$

If (γ_x, γ_y) is a unit vector so that $\gamma_x^2 + \gamma_y^2 = 1$, then the point on the line closest to the origin is $\ell(\gamma_x, \gamma_y)$, located at a distance ℓ from the origin. The slope of the line to that point is γ_y/γ_x : hence, the line from the origin to the closest point on $\gamma_x x + \gamma_y y = \ell$ is perpendicular to it. This fact will be used to determine the apothem d of the polygon.

Proceeding now towards the requirement to contain all ellipses inside the regular polygon, we find the maximum value of ℓ for which the side of the polygon intersects the ellipse. Our derivation for a polygon follows the first order Karush–Kuhn–Tucker conditions described by Kallrath and Rebennack [22] for a rectangular container.

For an ellipse to be contained inside the polygon, all sides $\gamma_x x + \gamma_y y$ must be less than or equal to this maximum value. Therefore we consider the following equation using λ as the maximizing embedded Lagrange multiplier.

$$\gamma_x x + \gamma_y y = \lambda \cdot e(a, b, x^c, y^c, \theta; x, y). \quad (3)$$

Differentiating both sides of equation (3) with respect to x , y , and λ , we obtain

$$\gamma_x x^c + \gamma_y y^c - \sqrt{\tau}, \quad (4)$$

$$\gamma_x x^c + \gamma_y y^c + \sqrt{\tau},$$

where

$$\begin{aligned} \tau = & (a^2 \gamma_x^2 + b^2 \gamma_y^2) \cos(\theta)^2 + (b^2 \gamma_x^2 + a^2 \gamma_y^2) \sin(\theta)^2 \\ & + (a - b)(a + b) \gamma_x \gamma_y \sin(2\theta). \end{aligned} \quad (5)$$

Notice that the Lagrange multiplier λ does not appear in (4). More importantly, the maximum value in the direction (γ_x, γ_y) is the result in (4) with the positive sign in front of $\sqrt{\tau}$. This explicit analytical result will be used in our optimization strategy for all sides of the regular polygon, which, of course, share the same apothem d .

Based on the above discussion, the condition of containing a given ellipse inside the polygon can be described by the relation

$$d \geq p(a, b, x^c, y^c, \theta; x, y) := \gamma_x x^c + \gamma_y y^c + \sqrt{\tau}. \quad (6)$$

In order to complete our analysis related to the sides of the container polygon, note that for a regular polygon with m sides, the points $(\gamma_{x_k}, \gamma_{y_k})$ that define the unit vectors for each apothem (i.e., for each side k of the polygon) are given by

$$(\gamma_{x_k}, \gamma_{y_k}) = \left(\cos\left(\frac{2 \cdot k \cdot \pi}{m} - \frac{\pi}{2}\right), \sin\left(\frac{2 \cdot k \cdot \pi}{m} - \frac{\pi}{2}\right) \right), k = 1, \dots, m. \quad (7)$$

Proceeding next towards preventing ellipse overlaps, it is useful to determine equations for the derivatives of the ellipse equation (1) with respect to x and y . For a given set of values (a, b, x^c, y^c, θ) we will denote $e(a, b, x^c, y^c, \theta; x, y)$ simply as $e(x, y)$: then these derivatives are determined as

$$\frac{de(x, y)}{dx} = \frac{2}{a^2 b^2} (b^2 (x - x^c) \cos(\theta)^2 - (a^2 - b^2)(y - y^c) \cos(\theta) \sin(\theta) + a^2 (x - x^c) \sin(\theta)^2), \quad (8)$$

$$\frac{de(x, y)}{dy} = \frac{2}{a^2 b^2} (a^2 (y - y^c) \cos(\theta)^2 - (a^2 - b^2)(x - x^c) \cos(\theta) \sin(\theta) + b^2 (y - y^c) \sin(\theta)^2). \quad (9)$$

All pairs of packed ellipses are prevented from overlapping by requiring that the minimum value of the ellipse equation for the first ellipse (ellipse i) for any point on the second ellipse (ellipse j) has to be greater than a judiciously set, sufficiently small parameter $\varepsilon \geq 0$. This requirement will be met using the embedded Lagrange multiplier method.

The next set of equations serves to 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. During optimization, this requirement with respect to the sign of λ will be enforced by setting its search bounds.

$$\frac{de(x, y)}{dx} (a_i, b_i, x_i^c, y_i^c, \theta_i; x, y) = \lambda \cdot \frac{de(x, y)}{dx} (a_j, b_j, x_j^c, y_j^c, \theta_j; x, y), \quad (10)$$

$$\begin{aligned}\frac{de(x,y)}{dy}(a_i, b_i, x_i^c, y_i^c, \theta_i; x, y) &= \lambda \cdot \frac{de(x,y)}{dy}(a_j, b_j, x_j^c, y_j^c, \theta_j; x, y), \\ e(a_j, b_j, x_j^c, y_j^c, \theta_j)(x, y) &= 0.\end{aligned}$$

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

$$\begin{aligned}\frac{de(x,y)}{dy}(a_i, b_i, x_i^c, y_i^c, \theta_i; x, y) \cdot \frac{de(x,y)}{dx}(a_j, b_j, x_j^c, y_j^c, \theta_j; x, y) &= \\ \frac{de(x,y)}{dy}(a_j, b_j, x_j^c, y_j^c, \theta_j; x, y) \cdot \frac{de(x,y)}{dx}(a_i, b_i, x_i^c, y_i^c, \theta_i; x, y).\end{aligned}\quad (11)$$

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

To illustrate the approach, consider the ellipses $e_i(1.25, 0.75, 1, 2, \pi/3; x, y)$ and $e_j(1.5, 0.83, -0.5, 1, \pi/4; x, y)$. The overlapping value of equation (1) for ellipse i between the ellipses is -0.886 with $(x, y) = (0.701, 1.777)$: this value can be found by solving the set of equations (10). Figure 2 shows the ellipse configuration.

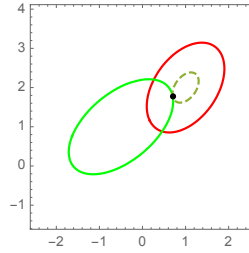


Figure 2. Two overlapping ellipses

To give another example, consider two ellipses defined by $e_i(1.25, 0.75, -1, -2, \pi/3; x, y)$ and $e_j(1.5, 0.83, -0.5, 1, \pi/4; x, y)$. The non-overlapping value between the ellipses is 1.84 with $(x, y) = (-0.758, -0.141)$: again, this value can be found by solving the set of equations (10) for x and y . Figure 3 shows the resulting non-overlapping ellipse configuration.

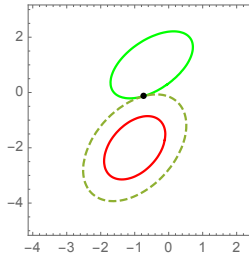


Figure 3. Two non-overlapping ellipses

Constructing Lagrange multiplier equations (10) for $e_i(x, y)$ and $e_j(x, y)$, as shown above, and requiring that the Lagrange multiplier be negative and the value of $e_i(x, y)$ be

positive, guarantees that these two ellipses do not overlap. Note that requiring $e_i(x, y) \geq 0$ allows the ellipses to touch. Consequently, the requirement that two ellipses do not overlap can be implemented by creating constraints from the Lagrange multiplier equations (10). That is, we require five constraints for each pair of ellipses: $\lambda_{i,j} \leq 0$ (a bound constraint), $e_i(x, y) \geq 0$, $e_j(x, y) = 0$, and the two Lagrange multiplier equations generated by differentiating the equation $e_i(x, y) = \lambda_{i,j} \cdot e_j(x, y)$ with respect to x and y . In the optimization framework, $\lambda_{i,j} \leq 0$ are the Lagrange multipliers appearing in the equations to find the point $(x_{j,i}, y_{j,i})$ on ellipse j that minimizes the value of the equation that describes ellipse i . Finally, we state constraints to prevent ellipse i from overlapping with ellipse j , by requiring that the minimal value of the equation describing ellipse i at the minimizing point on ellipse j has to be at least ε . We remark that the optimization solvers used in our study are not too sensitive to the choice of this parameter: for instance, values of $\varepsilon = 0, 10^{-6}$, and 10^{-2} have been all tested without leading to numerical issues.

Summarizing the model development steps, we obtain the following model-class for packing n general ellipses into an optimized regular m -polygon container.

$$\begin{aligned}
& \text{minimize} && d && (12) \\
& \text{subject to} && d \geq p(a_i, b_i, x_i^c, y_i^c, \theta_i; \gamma_{x_k}, \gamma_{y_k}) && \text{for } i = 1, \dots, n \\
& && && k = 1, \dots, m \\
& && \frac{de(x, y)}{dx}(a_i, b_i, x_i^c, y_i^c, \theta_i; x_{j,i}, y_{j,i}) && \text{for } i = 1, \dots, n-1 \\
& && && j = i+1, \dots, n \\
& && = \lambda_{j,i} \cdot \frac{de(x, y)}{dx}(a_j, b_j, x_j^c, y_j^c, \theta_j; x_{j,i}, y_{j,i}) \\
& && \frac{de(x, y)}{dy}(a_i, b_i, x_i^c, y_i^c, \theta_i; x_{j,i}, y_{j,i}) && \text{for } i = 1, \dots, n-1 \\
& && && j = i+1, \dots, n \\
& && = \lambda_{j,i} \cdot \frac{de(x, y)}{dy}(a_j, b_j, x_j^c, y_j^c, \theta_j; x_{j,i}, y_{j,i}) \\
& && e(a_j, b_j, x_j^c, y_j^c, \theta_j; x_{j,i}, y_{j,i}) = 0 && \text{for } i = 1, \dots, n-1 \\
& && && j = i+1, \dots, n \\
& && e(a_i, b_i, x_i^c, y_i^c, \theta_i) \geq \varepsilon && \text{for } i = 1, \dots, n-1 \\
& && && j = i+1, \dots, n \\
& && l \leq x_i^c \leq u && \text{for } i = 1, \dots, n \\
& && l \leq y_i^c \leq u && \text{for } i = 1, \dots, n \\
& && -\pi \leq \theta_i \leq \pi && \text{for } i = 1, \dots, n \\
& && l \leq x_{j,i} \leq u && \text{for } i = 1, \dots, n-1 \\
& && && j = i+1, \dots, n \\
& && l \leq y_{j,i} \leq u && \text{for } i = 1, \dots, n-1
\end{aligned}$$

$$\begin{aligned}
& j = i + 1, \dots, n \\
& 2 \cdot l \leq \lambda_{j,i} \leq 0 \qquad \text{for } i = 1, \dots, n-1 \\
& \qquad \qquad \qquad j = i + 1, \dots, n
\end{aligned}$$

In the model (12) l and u denote lower and upper bounds for the ellipse center positions: these bounds can be defined appropriately for each ellipse packing instance studied, in order to facilitate the finding of feasible solutions.

The optimization model (12) has $1 + 3n + 3(n-1)^2$ decision variables. In addition to the bound constraints imposed on all decision variables, the model has $m \cdot n + 4(n-1)^2$ non-convex constraints. Considering also the formulas introduced earlier for the ellipses, model (12) represents a hard global optimization challenge: both the number of decision variables and the number of nonconvex constraints increase as a quadratic function of n , observing also the term $m \cdot n$. For example, a problem instance with $n = 10$ ellipses inside a regular polygon with $m = 8$ sides leads to a model with 274 decision variables, corresponding bound constraints, and 404 non-convex constraints. Arguably, this rather small model instance can be already perceived as a computational global optimization challenge. Based on the above observations, we can expect that the computational difficulty of the model-class (12) rapidly increases as a function of n and m .

3 Numerical Global Optimization Applied to Ellipse Packing Problems

Considering even far less complicated object packing models than the model-class introduced here, one cannot expect to find general analytical solutions. Therefore we have been applying numerical global optimization to find high quality feasible numerical solutions to a range of challenging OPPs. Without going into details, we only refer here to some of our own related studies: consult e.g. Pintér [29], Stortelder *et al.* [48], Riskin *et al.* [46], Castillo and Sim [8], Pintér and Kampas [37, 38], Kampas and Pintér [23], Castillo *et al.* [7], Pintér and Kampas [39], Kampas *et al.* [24, 25], Pintér *et al.* [41]. Similarly to most of the works listed here, in the present study we utilize the Lipschitz Global Optimizer (LGO) solver system for global-local nonlinear optimization: here specifically, using its implementation linked to the computing system *Mathematica* [55], by Wolfram Research.

LGO (Pintér [34]) is aimed at finding the numerical global optimum of model instances from a very general class of global optimization problems. In order to use LGO within a theoretically rigorous framework, the key model assumptions are: bounded, non-empty feasible region; continuous or Lipschitz-continuous objective and constraint functions. These analytical conditions can be simply verified by inspection for model (12). LGO has been in use since the early 1990s, and it has been documented in detail elsewhere. In particular, Pintér [27] presents adaptive deterministic partition strategies and stochastic search methods, to solve global optimization problems under Lipschitz-continuity or mere continuity assumptions. The exhaustive search capability of these general algorithmic procedures guarantees their theoretical global convergence (correspondingly, in a deterministic sense or with probability 1).

The core solver system in LGO with implementations linked to various modeling platforms has been discussed e.g., by Pintér [27, 28, 30-33], and by Pintér *et al.* [42]. For

more recent development work, benchmarking studies and real-world applications, consult e.g. Çağlayan and Pintér [6], Deschaine *et al.* [11], Pintér and Horváth [36], Pintér and Kampas [39], Pintér [35]. Current implementation details are described in the LGO manual [34], with reference to a broad range of further LGO applications. LGO integrates several derivative-free global and local optimization strategies, without requiring higher-order model function information. These strategies include a sampling procedure as a global pre-solver, a branch-and-bound global search method (BB), global adaptive random search (RS), multi-start based global random search (MS), and local search (LS). According to extensive numerical experience, in difficult GO models, MS with subsequent LS solver phases often finds the best numerical solution. For this reason, the recommended default LGO solver option is MS + LS: this has been utilized also in our present numerical study.

The implementation of LGO with a link to *Mathematica* – with the software product name *MathOptimizer Professional* (Pintér and Kampas [40]) – has been extensively used in our benchmarking studies, and it has been used also here. *MathOptimizer Professional* users formulate their optimization model in *Mathematica*; this model is automatically translated into C or Fortran code; the converted model is solved by LGO; finally, the results are seamlessly returned to the calling *Mathematica* work document. The outlined structure supports the combination of *Mathematica*’s model development, visualization, and other capabilities with the robust performance and speed of the external LGO solver engine. LGO solver performance compares favorably not only to the related numerical optimization capabilities of *Mathematica*, but also to a broad range of other derivative-free nonlinear solvers. In support of the latter remark, consult e.g. the substantial benchmarking study of Rios and Sahinidis [45] in which, at the time of their study, a several years old LGO version has been one of the most efficient solvers. The current LGO release – as well as its *MathOptimizer Professional* implementation – supports the numerical solution of nonlinear optimization models with thousands of variables and general constraints. To illustrate this aspect, we present a substantial set of ellipse packing test results in the next section.

4 Illustrative Results

Our numerical experiments were mostly performed on a PC running under Windows 7, with an Intel Core i5 processor running at 2.6 GHz, with 16 GBytes of RAM, using *MathOptimizer Professional* running in *Mathematica* (versions 10 and 11), and using the *gcc* compiler [53] to automatically generate the necessary model input files for LGO. While we cannot guarantee the theoretical optimality of the ellipse configurations found, our computational results lead to visibly high quality packings. Let us also point out that *MathOptimizer Professional* is used here with its default parameter settings, without any “tweaking.”

To our best knowledge, there are no previously studied model instances available for the ellipse packing problem-class considered in our present work. Thus we first proceed to dealing with the basic setting presented in Honvault [18] of packing three congruent ellipses in a square: $n = 3$, all ellipses sharing the same eccentricity, and $m = 4$. In order to reproduce the results presented in Honvault [18], we use the following input structure: $a_i = 1$, $b_i = c$ for $i = 1, \dots, 3$, $c = 0.01, 0.02, \dots, 1.00$ is the eccentricity parameter of the ellipses. We note that Honvault [18] used a stochastic algorithm based on the so-called

inflation formula presented in Honvault [17]. Figure 4 numerically captures the function between the packing fraction and the eccentricity input parameter c . In this set of examples, solving each problem took less than 2 seconds of CPU time. Honvault [18] has shown analytically that for congruent ellipses placed inside a square the function between the packing fraction and the eccentricity is continuous, reaching the maximal packing fraction at an eccentricity of $1/3$, with optimum value of $\pi/4 \sim 0.7853981634$: this is comparable to the fraction **0.78089** obtained in our numerical results at $c = 0.33$, given our discretized eccentricity parameter structure. (We did not spend further effort on this rather simple illustrative example.)

Evidently, a regular polygonal container approaches a circular container as the number of sides m increases. Following the same input structure, Figure 4 captures the function between the packing fraction and the eccentricity for $m = 40$. In this case, the packing fraction reaches its approximate maximum of **0.86482** at eccentricity $c = 0.58$. For $m = 40$ solving each problem took less than 40 seconds.

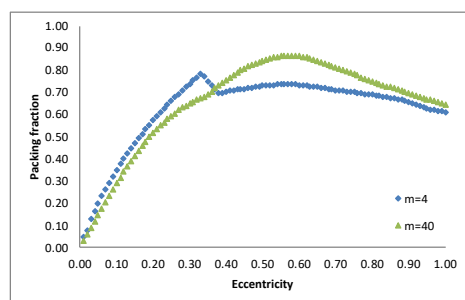


Figure 4. Packing fraction vs. eccentricity c for $n = 3$ congruent ellipses

Next, we consider packing ellipses into containers with *a priori* bounded size. The problem instances summarized in Table 1 are cited from Kallrath and Rebennack [22]. Table 2 includes computational results using the solver NMinimize, available in *Mathematica* and the solver BARON (Tawarmalani and Sahinidis [52]), available via the NEOS Server (<https://neos-server.org/neos/>). Specifically, for NMinimize we tested its Automatic, SimulatedAnnealing, and RandomSearch option settings. Among these, RandomSearch provided the most consistent results: hence, these results are reported in the table. Let us also note that BARON does not support trigonometric functions. In order to handle our model using BARON, we replaced function $\cos(\theta)$ with the decision variable Φ , function $\sin(\theta)$ with the decision variable Ψ , and added the constraint $\Phi^2 + \Psi^2 = 1$ with the bounds $-1 \leq \Phi, \Psi \leq 1$. All model instances were solved for a square container ($m = 4$) with the same overall lower and upper bound values. Our optimization strategy consistently provided superior results, both in terms of higher quality packings found and (typically) much lower CPU times.

Table 1. Ellipse packing instances from Kallrath and Rebennack [22]

Test case	(a_i, b_i)	Total area to be packed
TC2a	$(2.0, 1.5), (1.5, 1.0)$	14.13717

TC2b	(2.0,1.5), (1.8,1.4)	17.34159
TC3a	TC2a + (1.0,0.8)	16.65044
TC3b	TC2b + (0.8,0.7)	19.10088
TC4a	TC3a + (0.9,0.75)	18.77102
TC4b	TC3b + (1.1,1.0)	22.55664
TC5a	TC4a + (0.8,0.6)	20.27898
TC5b	TC4b + (0.9,0.8)	24.81858
TC6	TC5a + (0.7,0.3)	20.93872
TC11	(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
TC14	$7 \cdot (1.0,0.75) + 7 \cdot (0.5,0.375)$	20.6167

Table 3 summarizes additional computational results, including those from Kallrath and Rebennack [22] and Stoyan et al. [49]: these works were aimed at packing ellipses in optimized rectangles. Since our model can be extended to consider general convex polygons, our results include optimized regular polygonal containers with $m = 4$ (square), general polygonal containers with $m = 4$ (rectangle), and regular polygonal containers with $m = 8$ (octagon). The configuration geometries, as mentioned, are rather different given the different optimized containers: however, we gain an overall impression regarding the packing densities that can be achieved for a range of model instances. In terms of the quality of the packings for a rectangular container, our modeling and optimization strategy often matches (within 0.1% accuracy) or improves the best results previously reported. It is interesting to note that for test cases TC5b and TC6, the packing fractions with a regular octagonal container (our results) approach the packing fractions obtained using a rectangular container.

Next, we proceed to packing ellipses of (in principle) arbitrary size and orientation. The ellipse packing problem instances summarized in Table 4 have been generated with the following input structure: $a_i = 1/\sqrt{i}$, $b_i = a_i/c$ for $i = 1, \dots, n$, where c again denotes the eccentricity of the ellipses. Note that with $c = 1$ this problem becomes a general circle packing problem for circles with radii $a_i = 1/\sqrt{i}$. This choice guarantees that for the total area $A(n)$ of the packed ellipses we have the relation $A(n) \rightarrow \infty$ as $n \rightarrow \infty$. In our opinion, this general property can make the corresponding packing model instances more difficult to handle, as opposed to packing objects into containers with *a priori* bounded size.

Tables 5-11 summarize the computational results, for a total of 210 ellipse packing problems solved. In these model instances, the number of ellipses n is chosen as 3, 4, 5, 6, 8, or 10; for each given n , the eccentricity parameter c is chosen as one of the values 1.00, 1.05, 1.15, 1.35, 1.50, 1.75, or 2.00; and for each fixed pair (n, c) the number of regular polygon container sides m is chosen as one of 3, 4, 5, 6, or 8.

Table 2. Comparison of ellipse packing instances from Kallrath and Rebennack [22] with $m = 4$

Test case	NMinimize			BARON (via the NEOS Server)			Our result		
	Area of optimized container	Packing fraction	Time (sec.)	Area of optimized container	Packing fraction	Time (sec.)	Area of optimized container	Packing fraction	Time (sec.)
TC2a	22.66582	0.62372	4.09	22.66560	0.62373	1169.21	22.66580	0.62372	1.09
TC2b	29.33227	0.59121	2.80	29.33199	0.59122	1512.59	29.33230	0.59121	1.08
TC3a			^a	23.81605	0.69913	^c	22.66580	0.73461	1.53
TC3b			^a	29.45807	0.64841	^c	29.33230	0.65119	1.56
TC4a			^a	30.11192	0.62338	^c	25.51690	0.73563	3.32
TC4b	32.97942	0.68396	61.28	37.25437	0.60548	^c	29.37490	0.76789	3.17
TC5a			^a	33.15818	0.61158	^c	27.47430	0.73811	6.29
TC5b	34.37538	0.72199	1530.41	41.03620	0.60480	^c	32.87250	0.75500	8.16
TC6	27.97028	0.74861	2747.89	34.16182	0.61293	^c	26.28960	0.79646	22.01
TC11			^b			^b	60.85930	0.77741	762.84
TC14			^b	59.69047	0.34539	^c	28.91880	0.71292	2555.89

^a Infeasible result returned.

^b No result returned within 8hrs.

^c Best solution returned within 8hrs.

Table 3. Additional computational results for different configuration geometries

Test case	Rectangular container Kallrath and Rebennack [22]		Rectangular container Stoyan <i>et al.</i> [49]		General polygon with $m = 4$ (rectangle) Our result		Regular polygon with $m = 4$ (square) Our result		Regular polygon with $m = 8$ Our result		
	Area of optimized container	Packing fraction	Area of optimized container	Packing fraction	Area of optimized container	Packing fraction	Area of optimized container	Packing fraction	Area of optimized container	Packing fraction	
TC2a	18.00000	0.78540		a	18.00000	0.78540	b	22.66580	0.62372	20.52970	0.68862
TC2b	22.23152	0.78005		a	22.23160	0.78004	b	29.33230	0.59121	27.17870	0.63806
TC3a	21.38577	0.77858		a	21.38580	0.77858	b	22.66580	0.73461	21.18540	0.78594
TC3b	25.22467	0.75723		a	25.22470	0.75723	b	29.33230	0.65119	27.17870	0.70279
TC4a	23.18708	0.80955		a	23.18710	0.80955	b	25.51690	0.73563	23.80430	0.78855
TC4b	28.54159	0.79031		a	28.54090	0.79033	b	29.37490	0.76789	28.11220	0.80238
TC5a	25.29557	0.80168	24.55368	0.82590	24.56800	0.82542	b	27.47430	0.73811	25.19900	0.80475
TC5b	31.28873	0.79321	30.84870	0.80453	30.64920	0.80976	c	32.87250	0.75500	30.94290	0.80208
TC6	25.51043	0.82079	25.47173	0.82204	25.08330	0.83477	c	26.28960	0.79646	25.57010	0.81887
TC11	64.59177	0.73248	57.17830	0.82745	56.76690	0.83345	c,d	60.85930	0.77741	59.63610	0.79335
TC14	29.65886	0.69513	24.25099	0.85014	24.46450	0.84272	d	28.91880	0.71292	25.54360	0.80712

^a Not reported.^b Matches best result within 0.1% accuracy.^c Improves best result previously reported.^d Obtained using our “naïve” heuristic approach.

The CPU times are reasonable for the entire range of parameter choices, ranging from seconds to a few minutes. Note that these tables also include the packing fraction and the maximum constraint violation.

Table 4. Ellipse packing instances

Number of ellipses	Eccentricity	Total area to be packed
3	1.00	5.75959
4		6.54498
5		7.17330
6		7.69690
8		8.53840
10		9.20163
3	1.05	5.48532
4		6.23332
5		6.83172
6		7.33038
8		8.13181
10		8.76345
3	1.15	5.00834
4		5.69129
5		6.23765
6		6.69296
8		7.42470
10		8.00141
3	1.25	4.60767
4		5.23599
5		5.73864
6		6.15752
8		6.83072
10		7.36130
3	1.50	3.83972
4		4.36332
5		4.78220
6		5.13127
8		5.69227
10		6.13442
3	1.75	3.29119
4		3.73999
5		4.09903
6		4.39823
8		4.87909
10		5.25807
3	2.00	2.87979
4		3.27249
5		3.58665

6	3.84845
8	4.26920
10	4.60081

Table 5. Ellipse packing results for eccentricity $c = 1.00$

Problem number	Number of ellipses	Number of container sides	Area of optimized container	Packing fraction	Time (sec.)	Max constraint violation
1	3	3	9.31692	0.61819	2.92	4.59E-13
2		4	9.11142	0.63213	1.44	2.98E-10
3		5	9.04654	0.63666	1.59	1.12E-09
4		6	9.12255	0.63136	1.70	5.18E-10
5		8	9.28338	0.62042	2.09	9.89E-09
6	4	3	9.31692	0.70248	2.82	2.97E-09
7		4	9.31373	0.70272	3.06	5.86E-09
8		5	9.26474	0.70644	3.46	1.27E-13
9		6	9.56947	0.68394	3.76	1.30E-11
10		8	9.32100	0.70218	4.68	1.03E-12
11	5	3	11.10010	0.64624	6.83	2.35E-10
12		4	9.56677	0.74981	7.64	5.72E-12
13		5	10.12490	0.70848	8.50	3.12E-09
14		6	9.39010	0.76392	9.52	1.64E-14
15		8	9.93319	0.72216	12.01	5.39E-09
16	6	3	11.38690	0.67595	17.28	8.23E-09
17		4	10.64650	0.72295	19.19	4.05E-09
18		5	10.81850	0.71146	22.46	1.20E-09
19		6	10.82760	0.71086	24.71	1.39E-09
20		8	10.46650	0.73538	28.78	2.40E-12
21	8	3	12.74630	0.66987	87.67	8.70E-09
22		4	11.40820	0.74845	92.46	1.49E-12
23		5	11.72700	0.72810	109.47	3.55E-14
24		6	11.62580	0.73444	116.02	5.50E-09
25		8	11.87280	0.71916	130.20	9.43E-09
26	10	3	13.13800	0.70038	418.02	6.18E-09
27		4	12.47130	0.73782	334.19	8.33E-10
28		5	12.42860	0.74036	379.95	2.18E-13
29		6	12.65350	0.72720	386.65	5.17E-09
30		8	12.92340	0.71201	433.04	6.93E-10

Table 6. Ellipse packing results for eccentricity $c = 1.05$

Problem number	Number of ellipses	Number of container sides	Area of optimized container	Packing fraction	Time (sec.)	Max constraint violation
31	3	3	8.65035	0.63412	1.58	9.32E-09

32		4	8.40929	0.65229	1.47	4.39E-09
33		5	8.31090	0.66002	1.73	5.89E-09
34		6	8.40941	0.65228	1.86	9.97E-09
35		8	8.49902	0.64541	2.20	8.21E-09
36	4	3	8.65610	0.72011	2.96	9.95E-09
37		4	8.97455	0.69456	3.15	9.05E-09
38		5	8.48676	0.73448	3.96	9.77E-09
39		6	8.49949	0.73338	4.32	9.78E-09
40		8	8.59010	0.72564	5.41	8.84E-09
41	5	3	9.43556	0.72404	7.39	9.90E-09
42		4	8.94959	0.76336	8.03	8.98E-09
43		5	9.17626	0.74450	10.26	9.29E-09
44		6	9.51812	0.71776	11.08	5.46E-09
45		8	8.87133	0.77009	14.06	4.06E-09
46	6	3	10.62880	0.68967	18.80	8.53E-09
47		4	9.69256	0.75629	19.87	8.01E-09
48		5	9.76039	0.75103	24.57	6.40E-09
49		6	9.89178	0.74106	27.13	1.07E-11
50		8	9.76099	0.75099	33.87	9.97E-09
51	8	3	10.98320	0.74039	118.64	8.57E-09
52		4	10.61570	0.76602	96.45	5.06E-12
53		5	10.49460	0.77486	125.14	9.99E-09
54		6	11.36000	0.71583	125.25	9.56E-09
55		8	11.11580	0.73155	148.81	1.71E-09
56	10	3	12.36120	0.70895	302.30	1.46E-13
57		4	11.51620	0.76097	346.60	5.58E-10
58		5	11.28720	0.77641	441.08	7.87E-09
59		6	11.70320	0.74881	465.58	6.22E-09
60		8	11.30160	0.77542	302.47	6.45E-09

Table 7. Ellipse packing results for eccentricity $c = 1.15$

Problem number	Number of ellipses	Number of container sides	Area of optimized container	Packing fraction	Time (sec.)	Max constraint violation
61	3	3	7.55743	0.66270	1.64	8.97E-09
62		4	7.21585	0.69407	1.47	9.91E-09
63		5	7.10838	0.70457	1.75	9.66E-09
64		6	7.24809	0.69099	1.84	8.03E-09
65		8	7.19926	0.69567	2.25	7.99E-09
66	4	3	7.76378	0.73306	2.96	2.92E-10
67		4	7.43031	0.76596	3.15	6.68E-09
68		5	7.25170	0.78482	3.93	9.99E-09
69		6	7.43696	0.76527	4.32	9.17E-09
70		8	7.68404	0.74066	5.34	7.35E-09
71	5	3	8.24399	0.75663	7.47	4.02E-09
72		4	7.84139	0.79548	7.89	4.74E-09

73	6	5	8.00134	0.77958	10.05	3.99E-09
74		6	7.87724	0.79186	11.29	9.40E-09
75		8	8.01929	0.77783	13.93	9.77E-09
76		3	9.43381	0.70947	19.16	1.01E-10
77		4	8.64159	0.77451	20.20	5.36E-09
78		5	8.95285	0.74758	24.68	9.39E-09
79	8	6	8.50110	0.78731	28.53	6.63E-09
80		8	8.80568	0.76007	35.61	8.21E-09
81		3	10.06300	0.73782	98.19	4.94E-13
82		4	9.35476	0.79368	102.40	8.02E-09
83		5	9.63619	0.77050	120.98	9.91E-09
84		6	9.22623	0.80474	144.94	5.14E-09
85	10	8	9.61189	0.77245	158.59	8.94E-09
86		3	11.87970	0.67353	462.85	8.71E-09
87		4	10.17910	0.78607	384.85	9.97E-09
88		5	10.43410	0.76686	452.49	7.94E-10
89		6	10.28200	0.77820	494.91	1.52E-09
90		8	10.19040	0.78519	507.67	6.59E-09

Table 8. Ellipse packing results for eccentricity $c = 1.25$

Problem number	Number of ellipses	Number of container sides	Area of optimized container	Packing fraction	Time (sec.)	Max constraint violation
91	3	3	6.70414	0.68729	1.64	8.63E-09
92		4	6.34011	0.72675	1.48	9.78E-09
93		5	6.26118	0.73591	1.79	8.98E-09
94		6	6.35281	0.72530	1.86	6.64E-09
95		8	6.18144	0.74540	1.95	7.53E-09
96		3	7.03236	0.74456	3.00	8.43E-09
97	4	4	6.87038	0.76211	3.09	9.56E-09
98		5	6.57290	0.79660	3.96	8.90E-09
99		6	6.81065	0.76879	4.43	9.11E-09
100		8	6.84450	0.76499	5.46	9.33E-09
101		3	8.18732	0.70092	7.89	2.03E-09
102		4	7.42038	0.77336	8.52	4.68E-09
103	5	5	7.36284	0.77941	10.70	4.03E-09
104		6	7.59517	0.75557	11.22	6.71E-09
105		8	7.29897	0.78623	14.65	9.98E-09
106		3	8.17067	0.75361	19.59	9.91E-09
107		4	7.94288	0.77523	13.12	4.09E-14
108		5	7.77698	0.79176	28.30	8.35E-09
109	6	6	7.78206	0.79125	28.13	8.91E-09
110		8	7.70841	0.79881	36.44	1.98E-09
111		3	9.25429	0.73811	103.12	5.09E-09
112		4	8.95055	0.76316	112.29	8.54E-13
113		5	8.63591	0.79097	121.77	2.73E-09

114		6	8.45914	0.80750	153.32	8.00E-09
115		8	8.76421	0.77939	148.17	8.22E-10
116	10	3	9.60122	0.76670	406.82	1.08E-08
117		4	9.10444	0.80854	419.67	9.46E-09
118		5	8.95573	0.82197	505.96	5.22E-10
119		6	9.05680	0.81279	562.12	6.48E-09
120		8	9.11205	0.80787	550.35	6.28E-09

Table 9. Ellipse packing results for eccentricity $c = 1.50$

Problem number	Number of ellipses	Number of container sides	Area of optimized container	Packing fraction	Time (sec.)	Max constraint violation
121	3	3	5.46859	0.70214	1.64	6.52E-09
122		4	5.01028	0.76637	1.50	7.94E-09
123		5	4.92577	0.77952	1.78	8.29E-09
124		6	4.85285	0.79123	1.87	8.87E-09
125		8	4.70864	0.81546	2.20	5.66E-12
126	4	3	6.33790	0.68845	2.98	9.50E-09
127		4	5.69416	0.76628	3.24	1.65E-12
128		5	5.44826	0.80087	3.99	8.47E-09
129		6	5.46264	0.79876	4.63	8.69E-09
130		8	5.33806	0.81740	5.40	8.97E-09
131	5	3	6.73451	0.71010	4.95	9.96E-09
132		4	6.02743	0.79341	8.38	9.94E-09
133		5	5.87734	0.81367	8.61	9.90E-09
134		6	5.83059	0.82019	11.75	1.17E-10
135		8	5.96083	0.80227	14.02	8.97E-09
136	6	3	6.92542	0.74093	19.98	9.77E-09
137		4	6.27300	0.81799	21.29	2.91E-09
138		5	6.32142	0.81173	25.33	9.72E-09
139		6	6.14752	0.83469	28.88	6.43E-09
140		8	6.12832	0.83730	34.38	6.06E-11
141	8	3	7.35670	0.77375	118.69	6.95E-09
142		4	7.08329	0.80362	107.62	2.70E-11
143		5	7.04832	0.80761	130.10	8.17E-09
144		6	6.84629	0.83144	134.46	8.88E-09
145		8	6.89610	0.82543	183.64	9.36E-09
146	10	3	7.92676	0.77389	342.53	7.27E-10
147		4	7.64207	0.80272	495.94	7.85E-07
148		5	7.43587	0.82498	480.50	1.98E-09
149		6	7.42440	0.82625	579.58	1.33E-12
150		8	7.29268	0.84118	532.63	1.28E-09

Table 10. Ellipse packing results for eccentricity $c = 1.75$

Problem number	Number of ellipses	Number of container sides	Area of optimized container	Packing fraction	Time (sec.)	Max constraint violation
151	3	3	4.76150	0.69121	1.65	9.34E-09
152		4	4.16883	0.78948	1.50	9.16E-09
153		5	4.14857	0.79333	1.79	1.49E-12
154		6	4.01966	0.81877	1.93	3.79E-12
155		8	4.01870	0.81897	2.26	3.22E-12
156	4	3	4.96923	0.75263	2.81	2.52E-13
157		4	4.79465	0.78004	3.23	8.52E-09
158		5	4.49722	0.83162	3.90	9.08E-09
159		6	4.53262	0.82513	4.51	1.31E-10
160		8	4.47796	0.83520	5.44	9.85E-09
161	5	3	5.38494	0.76120	7.89	5.60E-11
162		4	5.21973	0.78530	8.27	9.97E-09
163		5	4.85210	0.84480	10.47	1.08E-12
164		6	4.92073	0.83301	12.17	9.83E-12
165		8	4.80767	0.85260	14.13	3.44E-09
166	6	3	5.68482	0.77368	20.28	7.24E-09
167		4	5.52502	0.79606	23.03	4.62E-09
168		5	5.30727	0.82872	26.36	7.65E-09
169		6	5.17500	0.84990	28.55	6.95E-09
170		8	5.15659	0.85293	34.80	6.11E-11
171	8	3	6.95629	0.70139	93.07	8.28E-09
172		4	5.91042	0.82551	99.22	8.56E-11
173		5	6.01020	0.81180	146.64	9.70E-10
174		6	5.88691	0.82880	142.68	8.88E-10
175		8	5.86351	0.83211	148.20	7.63E-09
176	10	3	6.62406	0.79378	371.81	5.45E-10
177		4	6.72522	0.78184	360.31	4.30E-10
178		5	6.45369	0.81474	476.52	2.60E-09
179		6	6.37592	0.82468	451.54	4.08E-10
180		8	6.24001	0.84264	532.80	4.29E-09

Table 11. Ellipse packing results for eccentricity $c = 2.00$

Problem number	Number of ellipses	Number of container sides	Area of optimized container	Packing fraction	Time (sec.)	Max constraint violation
181	3	3	4.26838	0.67468	1.64	7.46E-09
182		4	3.72254	0.77361	1.48	9.57E-11
183		5	3.72500	0.77310	1.76	2.02E-13
184		6	3.63638	0.79194	1.70	3.36E-09
185		8	3.56619	0.80753	2.18	1.84E-11
186	4	3	4.51450	0.72488	3.04	9.93E-09
187		4	4.13910	0.79063	3.28	5.66E-09

188		5	4.17102	0.78458	4.24	8.65E-09
189		6	4.10698	0.79681	4.32	1.37E-11
190		8	4.01400	0.81527	5.49	9.10E-12
191	5	3	4.70587	0.76217	7.22	3.37E-13
192		4	4.51516	0.79436	8.44	7.36E-09
193		5	4.35598	0.82339	10.51	1.35E-09
194		6	4.30684	0.83278	11.76	7.18E-13
195		8	4.24230	0.84545	14.48	7.93E-10
196	6	3	4.88883	0.78719	21.86	1.26E-12
197		4	4.86759	0.79063	21.95	3.72E-09
198		5	4.66241	0.82542	27.20	1.31E-09
199		6	4.61147	0.83454	28.83	6.25E-13
200		8	4.57293	0.84157	35.96	8.82E-09
201	8	3	6.03076	0.70790	96.61	7.12E-10
202		4	5.48223	0.77874	115.36	6.31E-09
203		5	5.14148	0.83035	106.28	1.98E-12
204		6	5.16038	0.82730	162.65	7.86E-09
205		8	5.12505	0.83301	163.99	8.93E-09
206	10	3	6.28063	0.73254	393.10	3.04E-09
207		4	5.77645	0.79648	338.82	1.08E-09
208		5	5.77305	0.79695	424.71	3.77E-09
209		6	5.60578	0.82073	449.17	1.13E-09
210		8	5.56694	0.82645	645.70	5.24E-10

Two illustrative packing configurations are displayed by Figures 5 and 6. The points shown on the ellipses (in light green color in the online version of this article) are the auxiliary points used for preventing ellipse overlap: note that for preventing the overlap of a pair of ellipses, there is a point on one of the two ellipses, not on both.

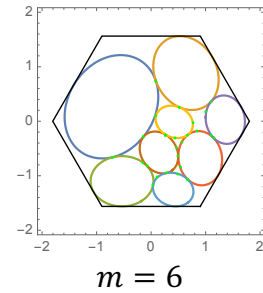
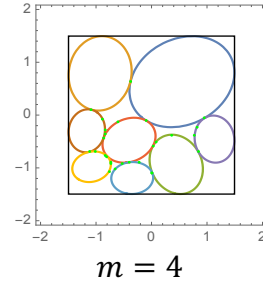
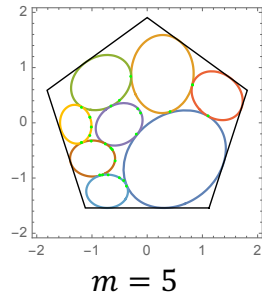
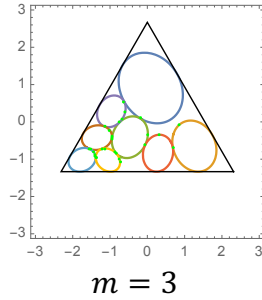


Figure 5. Packing configurations found for $n = 8$, $c = 1.25$, and $m = 3, 4, 5, 6$

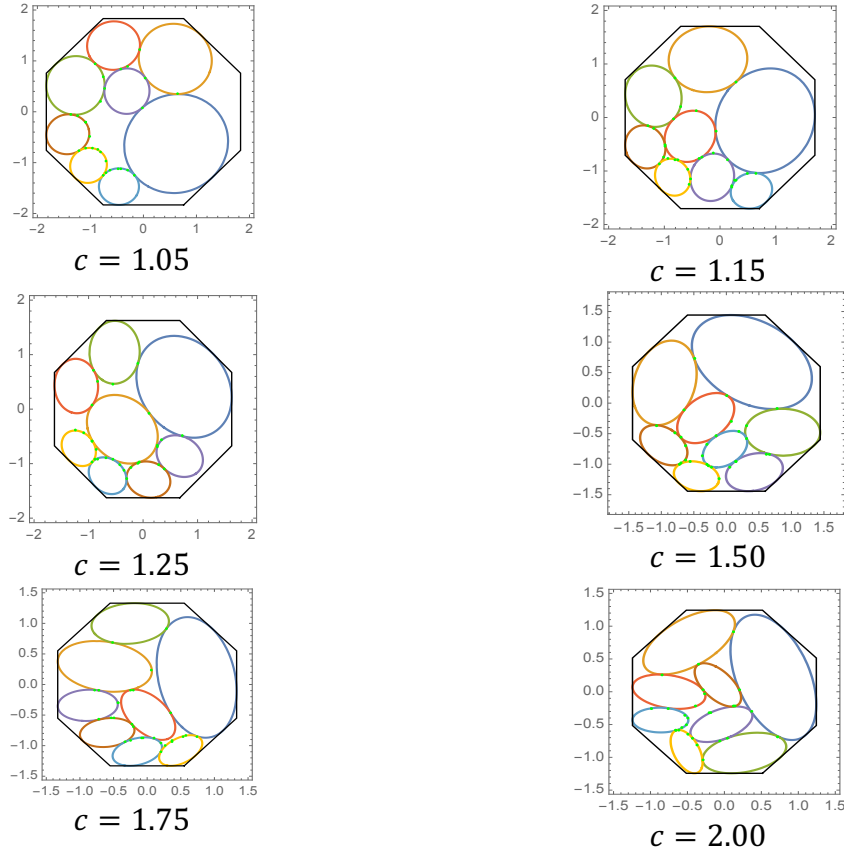


Figure 6. Packing configurations found for $n = 8$, $m = 8$, for all $c > 1.00$

Figures 7-9 capture the positive correlation between the packing fraction and the different input parameters: number of ellipses, eccentricity, and number of container sides. Multiple linear regression analysis indicates that the regression function for the packing fraction can be estimated as $0.5443 + 0.0068 \cdot n + 0.0893 \cdot c + 0.0104 \cdot m$, with p -values (i.e., observed significance levels, cf. Black *et al.* [5]) below 0.0001 for all input parameters. This finding indicates that we have very strong statistical evidence suggesting that the regression coefficients are different from zero. Figure 10 illustrates the quality of the regression equation by depicting actual vs. predicted packing fractions.

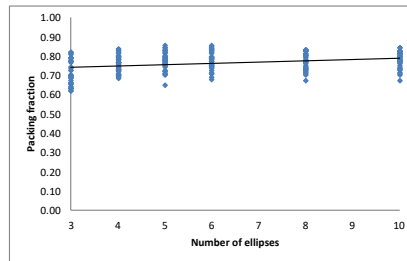


Figure 7. Packing fraction vs. number of ellipses n

(Diamonds represent the problem instances)

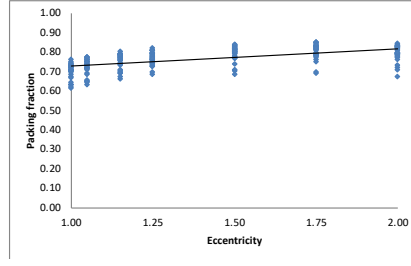


Figure 8. Packing fraction vs. eccentricity c
(Diamonds represent the problem instances)

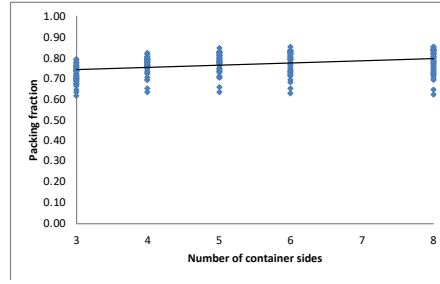


Figure 9. Packing fraction vs. number of container sides m
(Diamonds represent the problem instances)

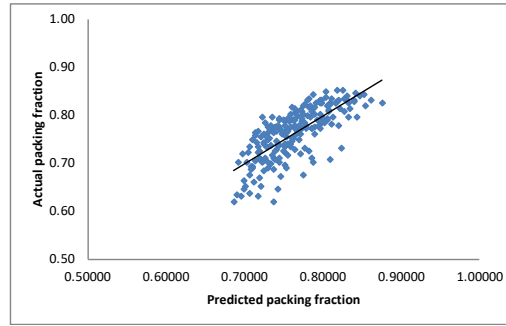


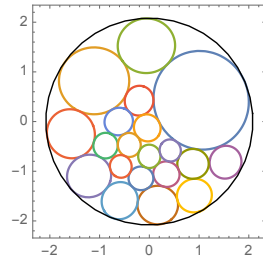
Figure 10. Actual (computed) *vs.* predicted packing fractions

To conclude the discussion of numerical results, let us point out that – based on our current model formulation and the experiments conducted – it is apparent that our default global optimization strategy could become prohibitively expensive for arbitrarily increasing m and n . Therefore, also based on related earlier research experience, we believe that large-scale problem instances will benefit from exploiting suitable, problem-dependent heuristic approaches. As a proof-of-concept, in Table 12 we present illustrative results using the input structure $a_i = 1/\sqrt{i}$, $b_i = a_i/c$, based on a random starting solution used by a single call to the local nonlinear optimization solver Ipopt. Ipopt is developed and maintained as part of the COIN-OR Initiative; consult <http://www.coin-or.org>. Our admittedly very simple heuristic strategy already gives promising results which certainly could be improved

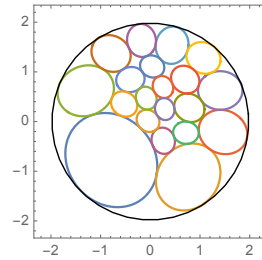
by using more sophisticated solution strategies. Figure 11 shows some of the packing configurations found for $n = 20$ and $m = 40$, using various eccentricity parameters.

Table 12. Ellipse packing results obtained using a heuristic approach for $m = 40$

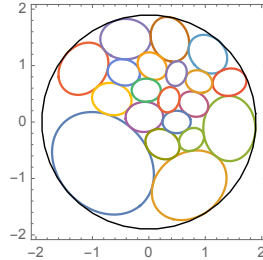
Problem number	Eccentricity	Number of ellipses	Total area to be packed	Area of optimized container	Packing fraction	Time (sec.)
211	1.00	10	9.20163	12.7618	0.72103	22.75
212		15	10.42452	13.9156	0.74913	162.81
213		20	11.30263	14.8480	0.76122	708.19
214	1.05	10	8.76345	11.5416	0.75929	24.25
215		15	9.92812	12.9380	0.76736	160.64
216		20	10.76441	13.5649	0.79355	744.36
217	1.15	10	8.00141	10.1715	0.78665	23.41
218		15	9.06480	11.4132	0.79424	157.95
219		20	9.82838	12.2667	0.80122	730.81
220	1.25	10	7.36130	9.2555	0.79534	23.34
221		15	8.33962	10.1671	0.82026	157.14
222		20	9.04211	11.2711	0.80224	726.80
223	1.50	10	6.13442	7.2432	0.84692	22.97
224		15	6.94968	8.3826	0.82906	158.47
225		20	7.53509	9.1134	0.82682	732.96
226	1.75	10	5.25807	6.3626	0.82641	23.50
227		15	5.95687	6.9961	0.85145	156.49
228		20	6.45865	7.7164	0.83700	722.20
229	2.00	10	4.60081	5.5010	0.83500	22.72
230		15	5.21226	6.4678	0.80588	158.05
231		20	5.65132	6.5778	0.85915	736.06



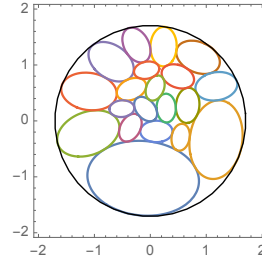
$c = 1.05$



$c = 1.15$



$c = 1.25$



$c = 1.50$



Figure 11. Packing configurations for $n = 20$ and $m = 40$, for all $c > 1.00$

5 Summary and Conclusions

In this work, we introduce the problem-class of packing, without overlap, n general ellipses into an optimized regular m -polygon. The packed ellipses can have arbitrary size, shape, and orientation. These packing problems lead to hard global optimization challenges: the number of model variables and general constraints are both rapidly increasing functions of m and n . The model formulation based on regular polygons also supports the approximation of different container shapes. Our model development approach can be directly extended to consider also general convex, but not necessarily regular, polygons.

The optimization strategy presented requires the determination of the maximal distance from the center of the polygon faces to each ellipse, in order to determine the size of the polygon's apothem, which is then minimized. To meet this requirement, we present explicit analytical formulas. Our strategy also requires the finding of the minimal distance between all pairs of the ellipses, in order to prevent the ellipses from overlapping. To satisfy the latter requirement, we develop embedded optimization calculations using Lagrange multipliers. In this framework, the optimization strategy proceeds simultaneously to find the optimized area of the regular polygon and to prevent ellipse overlaps. The use of embedded Lagrange multipliers allows us to determine quantities that cannot be found by explicitly formulated relations. To illustrate this point, see Figures 12 and 13 below with further object packing examples that are part of our closely related studies. These examples demonstrate the packing of circles, ellipses, and hyper-ellipses simultaneously into a regular polygon (Figure 12), and packing ellipses into a non-regular, elongated container polygon (Figure 13).

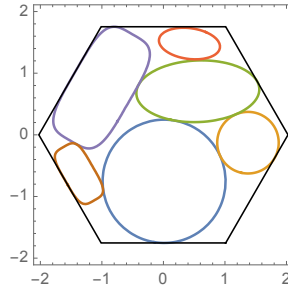


Figure 12. Packing example with circles, ellipses, and hyper-ellipses

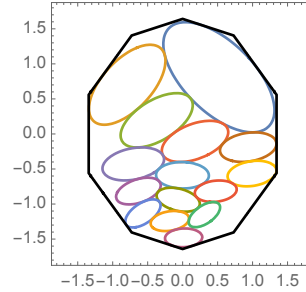


Figure 13. Packing example with ellipses in a non-regular container polygon

To solve the resulting highly non-convex optimization problems, we use the LGO solver system in its implementation linked to *Mathematica*. Our numerical results – including both global optimization and a simple random start + local search based “naïve” approach – successfully reproduce or surpass all previously published analytical results, in all examples considered here. Compared to several other optimization approaches, our optimization strategy consistently provides high quality packings, typically at a fraction of the computational time required by all other software used in our tests. These facts demonstrate the viability and efficiency of our modeling and optimization approach, for a wide range of parameter choices.

As a closing remark, we conjecture that regression analysis based on the results obtained supports the determination of statistically established estimates of the packing fraction in new model-instances from given model-classes. This observation is also consistently supported by our results across all model-classes considered here.

References

1. Alder, B.J., Wainwright, T.E., 1957. Phase transition for a hard sphere system. *Journal of Chemical Physics* 27, 1208-1209.
2. Bernal, J.D., 1959. Geometrical approach to the structure of liquids. *Nature* 183, 141-147.
3. Birgin, E.G., Lobato, R.D., Martínez, J.M., 2015. Packing ellipsoids by nonlinear optimization. *Journal of Global Optimization* 10.1007/s10898-015-0395-z, 1-35.
4. Birgin, E.G., Lobato, R.D., Martínez, J.M., 2016. A nonlinear programming model with implicit variables for packing ellipsoids. *Journal of Global Optimization* 10.1007/s10898-016-0483-8.
5. Black, K., Chakrapani, C., Castillo, I., 2014. *Business Statistics for Contemporary Decision Making* (2nd Edition). Wiley & Sons Canada, Toronto, ON, Canada.
6. 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.
7. 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.
8. Castillo, I., Sim, T., 2004. A spring-embedding approach for the facility layout problem. *Journal of the Operational Research Society* 55, 73-81.

9. Chaikin, P.M., 2000. Thermodynamics and hydrodynamics of hard spheres: the role of gravity. In: Cates, M.E., Evans, M.R., Eds. *Soft and Fragile Matter: Nonequilibrium Dynamics, Metastability and Flow*, Vol. 53, 315-348. Institute of Physics Publishing, Bristol.
10. Cheng, Z.D., Russell, W.B., Chaikin, P.M., 1999. Controlled growth of hard-sphere colloidal crystals. *Nature* 401, 893-895.
11. Deschaine, L.M., Lillys, T.P., Pintér, J.D., 2013. Groundwater remediation design using physics-based flow, transport, and optimization technologies. *Environmental Systems Research* 2:6.
<http://www.environmentalsystemsresearch.com/content/2/1/6>.
12. Edwards, S.F., 1994. The role of entropy in the specification of a powder. In: Mehta, A., Ed. *Granular Matter: An Interdisciplinary Approach*, pp. 121-140. Springer, New York.
13. 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.
14. Gensane, T., Honvault, P., 2014. Optimal packings of two ellipses in a square. *Forum Geometricorum* 14, 371-380.
15. Giachetti, R.E., Sanchez, J.C., 2009. A model to design recreational boat mooring fields. *Naval Research Logistics* 56, 158-174.
16. Hifi, M., M'Hallah, R., 2009. A literature review on circle and sphere packing problems: models and methodologies. *Advances in Operations Research*, DOI: 10.1155/2009/150624.
17. Honvault, P., 2011. Maximal inflation of two ellipses. *Technical Report. LMPA, Université du Littoral Côte d'Opale*, Calais Cedex, France.
18. Honvault, P., 2015. Density of optimal packings of three ellipses in a square. *Journal for Geometry and Graphics* 19, 201-209.
19. Jaeger, H.M., Nagel, S.R., 1992. Physics of the granular state. *Science* 255, 1523-1531.
20. Jaeger, H.M., Nagel, S.R., Behringer, R.P., 1996. Granular solids, liquids, and gases. *Reviews of Modern Physics* 68, 1259-1273.
21. Kallrath, J., 2017. Packing ellipsoids into volume-minimizing rectangular boxes. *Journal of Global Optimization* 67, 151-185.
22. Kallrath, J., Rebennack, S., 2014. Cutting ellipses from area-minimizing rectangles. *Journal of Global Optimization* 59, 405-437.
23. Kampas, F.J., Pintér, J.D., 2006. Configuration analysis and design by using optimization tools in *Mathematica*. *The Mathematica Journal* 10, 128-154.
24. Kampas, F.J., Pintér, J.D., Castillo, I., 2016. General ellipse packings in an optimized circle using embedded Lagrange multipliers. (Submitted) Available at http://www.optimization-online.org/DB_HTML/2016/01/5293.html.
25. Kampas, F.J., Pintér, J.D., Castillo, I., 2017. Optimal packing of general ellipses in a circle. In: Takáč, M. and Terlaky, T., Eds., *Modeling and Optimization: Theory and Applications (MOPTA 2016 Proceedings)*, pp. 23-38. Springer International Publishing AG, Cham, Switzerland.
26. 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.
27. Pintér, J.D., 1996. *Global Optimization in Action*. Kluwer Academic Publishers, Dordrecht.
28. 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.

29. Pintér, J.D., 2001. Globally optimized spherical point arrangements: model variants and illustrative results. *Annals of Operations Research* 104, 213-230.
30. 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.
31. 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.
32. Pintér, J.D., 2007. Nonlinear optimization with GAMS/LGO. *Journal of Global Optimization* 38, 79-101.
33. Pintér, J.D., 2009. Software development for global optimization. In: Pardalos, P.M., Coleman, T.F., Eds. *Global Optimization: Methods and Applications*, pp. 183-204. *Fields Institute Communications Volume 55*. Published by the American Mathematical Society, Providence, RI.
34. Pintér, J.D., 2016. *LGO – A Model Development and Solver System for Global-Local Nonlinear Optimization, User's Guide*. Published and distributed by PCS Inc., Canada.
35. Pintér, J.D., 2017. How difficult is nonlinear optimization? A practical solver tuning approach, with illustrative results. *Annals of Operations Research*, DOI: 10.1007/s10479-017-2518-z.
36. 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.
37. Pintér, J.D., Kampas, F.J., 2005. Nonlinear optimization in *Mathematica* with *MathOptimizer Professional*. *Mathematica in Education and Research* 10 (1), 1-18.
38. 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*, pp. 263-280. Springer Science + Business Media, New York.
39. 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.
40. Pintér, J.D., Kampas, F.J., 2015. *Getting Started with MathOptimizer Professional*. Published and distributed by PCS Inc., Canada.
41. Pintér, J.D., Kampas, F.J., Castillo, I., 2017. Globally optimized packings of non-uniform size spheres in \mathbb{R}^d : a computational study. *Optimization Letters*, DOI: 10.1007/s11590-017-1194-x.
42. Pintér, J.D., Linder, D., Chin, P., 2006. Global Optimization Toolbox for Maple: An introduction with illustrative applications. *Optimization Methods and Software* 21, 565-582.
43. Pusey, P.N., 1991. Colloidal suspensions. In: Hansen, J.P., Levesque, D., Zinnjustin, J., Eds. *Liquids, Freezing and Glass Transition*, Vol. 51 of Les Houches Summer School Session, pp. 763-942. Elsevier Science Publishers, Amsterdam.
44. Rintoul, M.D., Torquato, S., 1996. Metastability and crystallization in hard-sphere systems. *Physical Review Letters* 77, 4198-4201.
45. Rios, M.L., Sahinidis, 2013. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization* 56, 1247-1293.

46. Riskin, M.D., Bessette, K.C., Castillo, I., 2003. A logarithmic barrier approach to solving the dashboard planning problem. *INFOR* 41, 245-257.
47. Specht, E., 2018. <http://www.packomania.com/>. (Last update shown at website: June 29, 2018 as of December 12, 2018.)
48. 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.
49. Stoyan, Y., Pankratov, A., Romanova, T., 2016. Quasi-phi-functions and optimal packing of ellipses. *Journal of Global Optimization* 65, 283-307.
50. Stoyan, Y., Romanova, T., Pankratov, A., Chugay, A., 2015. Optimized object packings using quasi-phi functions. In: Fasano, G., Pintér, J.D., Eds. *Optimized Packing with Applications*, pp. 265-293. Springer Science + Business Media, New York.
51. Szabó, P.G., Markót, M.Cs, 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.
52. Tawarmalani, M. Sahinidis, N.V., 2005. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming* 103, 225-249.
53. The GNU Compiler Collection, 2016. <https://gcc.gnu.org/>.
54. Uhler, C., Wright, S.J., 2013. Packing ellipsoids with overlap. *SIAM Review* 55, 671-706.
55. Wolfram Research, 2018. *Mathematica* (Release 11). Wolfram Research, Inc., Champaign, IL.